AD-A096 678 AKRON UNIV OH COLL OF ENGINEERING F/G 20/11
EVALUATION OF ADINA. PART I. THEORY AND PROGRAMING DESCRIPTIONS--ETC(U)
JUN 80 T Y CHANG, J PADOVAN N00014-78-C-0691
UNCLASSIFIED AUE-801 NL

1 OF 2
AD A096678

LEVEL (12)

Report No. AUE-801

AD A096678

EVALUATION OF ADINA: Part I
Theory and Programming Descriptions

T. Y. Chang
J. Padovan

College of Engineering
The University of Akron
Akron, OH 44325

June 8, 1980

Final Report

Prepared for
Office of Naval Research
Department of the Navy
Contract N00014-78-C-0691

THE UNIVERSITY OF AKRON
FIAT LUX
1870

(10)

Report No. AUE-801

6

EVALUATION OF ADINA. PART I.

Theory and Programing Descriptions

10

T. Y. Chang

J. Padovan

College of Engineering
The University of Akron
Akron, Ohio 44325

11 8 Jun[redacted]80

12) 135

9 Final Report

Prepared for
Office of Naval Research
Department of the Navy

Contract N00014-78-C-0691

15

41 ... 11

Table of Contents

I. INTRODUCTION

During the last two decades the growth of finite element software for structure applications has experienced a vast changing environment. In the 1960's research efforts in finite element analysis were primarily concentrated on the development of discretization techniques for various structural geometrics. In this connection, numerious special purpose research programs were written for testing of the related numerical algorithms. Later, efforts were evolved into the development of large scale general purpose computer programs such as NASTRAN, STRUDL, etc. In the 1970's attentions were turned to the applications of the finite element method to nonlinear problems. These include understanding the nature of various nonlinear systems, investigations of nonlinear solution algorithms and development of both special and general purpose programs such as HONDO, PLANS, NONSAP and ADINA. Since nonlinearities in a structural system may have a wide range of meaning, it is difficult to write a software in complete generality, which can cover all the possible nonlinear features that a structure may encounter. Instead, most finite element programs were written with a limited scope; new features and solution methods are added as they become available.

Since tremendous amount of manpower and financial resources have already been invested in the development of various large scale general purpose finite element software, a significant cost is further necessary in order to maintain and upgrade these packages coping with the new computational technology and changing hardware configurations. Due to the high cost associated with the program maintenance and potential budget limitations, there is an apparent need to conduct a coordinated effort for further software development and maintenance. Consequently, several user's groups, e.g., SAP, STRUDL, ADINA, etc., have been formed for information and cost sharing. Also recognizing such a need, an ISEG (Interagency Software Evaluation Group) associated with the armed forces and various government agencies was formed for software user's coordination. The immediate goal of ISEG is to identify application software packages and perform in-depth evaluation of these packages. Initially, such effort is concentrated on the structural mechanics software due to the advanced status in this area. The selection criteria and evaluation procedure have been proposed by Nickell [1]. As a part of the ISEG effort, a nonlinear finite element program called ADINA [2] was selected for the software evaluation.

Among many nonlinear finite element programs that are available, ADINA represents one of the several which have been used extensively by government agencies and private companies. This program is an extended version of another nonlinear code, called NONSAP, which was developed at the University of California, Berkeley. Later, NONSAP was extended and modified by Bathe of MIT, one of the original NONSAP developers, to include additional features such as out-of-core solution scheme, double precision arithmetic, expand element and material libraries, etc. The extended version was then renamed as ADINA, (Automatic Dynamic Incremental Nonlinear Analysis).

Use of nonlinear finite element computer programs can be a difficult task due to the numerical complexity of nonlinear analysis in general and the limitation of the program in particular. When an analysis is performed, computation may be interrupted for a number of reasons and it is hard to pin-point the exact cause. In many cases, unsuccessful runs were due to the mis-use of the program by the analyst resulting from inadequate experience or the lack of proper understanding the limitations of the code. Any supportive information leading to the effective usage of the nonlinear program would be helpful from the user's standpoint. It is therefore the intent of this research effort to conduct a systematic review and evaluation of ADINA in terms of its capability and limitations for solving nonlinear structural problems.

By following the procedure outlined in [1], the evaluation work consists of the following items:

1) Theoretical basis, assumptions and numerical approaches adopted in the program.
2) Program architecture in terms of program design, use of control variables and data structure.
3) Review of nonlinear static and dynamic solution algorithms.
4) Identifications of typical algorithmic limitations encountered for both the nonlinear static and dynamic analyses.

Comments for the first two items are enumerated in Part I of this report, whereas the results obtained for the last two items are discussed in Part II.

Since at the beginning of this work, the 1977 ADINA was made available by the developer to the investigators. Therefore, all the comments and results are limited to this version of ADINA. Although a more recent version, 1978 ADINA*, was released, it was not made available for carrying out this study. Nevertheless, a description on the 1978 ADINA is outlined in section II of this report based on the revised user's manual.

---

* The 1978 ADINA was released to the ADINA user group in the later part of 1979.

II. GENERAL DESCRIPTION

ADINA [2,3] is a general purpose finite element program which was developed for conducting both linear and nonlinear, static and dynamic analysis for a range of structures. The program was extended from another nonlinear structural analysis program, i.e. its predecessor NONSAP [4], which was released to the public domain in 1974.

Although ADINA can be used for both linear and nonlinear structural analysis, the major function of the program is its unique capability in performing nonlinear analysis. Therefore, it is viewed primarily as a nonlinear code compared to other linear codes such as NASTRAN [5], STRUDL [6] and SAP [7].

The analysis capability of a general purpose finite element program may be best described by examining four different areas, namely

1) Analysis procedures
2) Element library
3) Material library
4) Other features

The above items for the 1977 and 1978 versions of ADINA in comparison with NONSAP are summarized in Table 1. Each of these areas is discussed in the following sections in accordance with the ADINA user's manual [2].

1. Analysis Procedures

In addition to its linear analysis capability, the analysis procedures consist of three major areas:

1) Nonlinear static analysis
2) Nonlinear dynamic analysis
3) Frequency analysis

Nonlinearities include both nonlinear materials and large deformations. However, the large deformation that can be utilized is limited to large displacement (or rotation) but small strain. This is due to the fact that the stress and strain measures adopted in the constitutive formulation of the code are not in compliance with the requirements of large strain analysis. A further discussion of this point will be given in section III.2. The solution method for solving nonlinear problems (both static and dynamic) is based on the incremental approach together with equilibrium iterations. For each load (or time) increment, tangent stiffness is used for solving the displacement equilibrium equations and the stiffness of the structure remains unchanged during iterations. Therefore, the method is equivalent to the well-known modified Newton-Raphson scheme for solving nonlinear equations by numerical approach [8].

For dynamic analysis, the equilibrium equations of the finite element system are solved by step-by-step numerical integration in time, for which both the implicit and explicit integration (or central difference) methods are adopted in the program.

Table 1 Summary of Analysis Options and Comparison with NONSAP

	1978 ADINA	1977 ADINA	NONSAP
Analysis Procedures			
Nonlinear Static Analysis	X	X	X
Nonlinear Dynamic Analysis:			
Implicit integration method	X	X	X
Explicit integration method	X	X	
Frequency Analysis	X	X	X
Element and Material Library			
Truss Element:	X	X	X
Linear elastic	X	X	X
Nonlinear elastic	X	X	X
Thermal-elastic	X	X	
Elastic-plastic	X	X	
Thermal-elastic-plastic-creep	X	X	
2/D Continuum	X	X	X
Isotropic linear elastic	X	X	X
Orthotropic linear elastic	X	X	X
Isotropic thermal-elastic	X	X	
Curve description model	X	X	X
Concrete model	X	X	
Elastic-plastic Drucker-Prager	X	X	X
Elastic-plastic Von Mises	X	X	X
Thermal-elastic-plastic-creep	X	X	
Mooney-Rivlin material	X	X	X
3/D Continuum	X	X	X
Isotropic linear elastic	X	X	X
Orthotropic linear elastic	X	X	
Isotropic thermal-elastic	X	X	
Curve description model	X	X	X
Concrete model	X	X	
Elastic-plastic Drucker-Prager	X		
Elastic-plastic Von Mises	X	X	
Thermal-elastic-plastic-creep	X	X	

	1978 ADINA	1977 ADINA	NONSAP
Element and Material Library			
Beam Element	X	X	
Linear elastic	X	X	
Elastic-plastic	X	X	
Shell Element	X		
Linear elastic	X		
Elastic-plastic	X		
2/D Fluid	X		
(Invicid compressible fluid with constant bulk modulus)			
3/D Fluid	X		
(Invicid Compressible fluid with constant bulk modulus)			
Other Features			
Out-of-core linear equation solver	X	X	
Analysis restart	X	X	X
Data check	X	X	X
Nodal force Loading	X	X	X
Pressure loading*	X	X	
Thermal loading	X	X	
Displacement boundary Conditions	X	X	
Companion thermal analysis code	X	X	
Data Card image printout	X	X	
Substructuring	X		

*Effect of geometric change was not included.

In the implicit integration, either Nemark's β-method or Wilson θ-method can be applied. Options are available to generate either the lumped mass or consistent mass matrix for the structure in the same way as a linear problem. However, if the central difference method is used, only the lumped mass matrix is permitted. In addition, concentrated masses can be added at selected degrees of freedom. To consider the damping effect, only concentrated nodal dampers are included and this information must be specified for each direction of those nodes having the damping effect.

In the frequency analysis, two solution algorithms, namely the determinant search method and the subspace iteration technique, are available. From the analysis, the lowest p eigenvalue and corresponding mode shapes are calculated. An over-relaxation scheme with a shifting strategy is also available in the subspace iteration technique to accelerate the convergence of the solution. Since the eigen-solution in ADINA is performed outside the time-integration loop, the stiffness matrix used in the analysis is' essentially linearly elastic even though the problem in question may be nonlinear. The mass matrix can be either lumped or consistent. No damping effect is considered in the frequency analysis.

As pointed out in the ADINA manual, the frequency analysis is desirable for estimating the resonance conditions of a vibrating system, or for selecting a suitable time step Δt in a direct step-by-step numerical integration. However, such estimation is valid only for a linear system.

2. Element Library

Element type is used to represent the geometric configuration of a specific structure. Therefore, the structural geometry that can be handled by a program is determined by the types of element available in the code. In the 1977 version of ADINA, there are basically four types of elements:

1) Three-dimensional truss
2) Three-dimensional beam
3) Two-dimensional (2-D) isoparametric solid (for plane stress, plane strain and axisymmetric deformations) with 4 to 8 nodes.
4) Three-dimensional (3-D) isoparametric solid with 8 to 20 nodes.

For the truss and beam elements, large displacement with small strain was assumed and the element stiffnesses were derived on the basis of the updated Lagrangian formulation (UL)*. Whereas for the 2-D and 3-D solid elements, both the total Lagrangian (TL) and' updated Lagrangian formulations* are available. Although a structure can be modeled by a combination of different element types, cautions must be given in two accounts:

1) In the case of large deformation analysis, either the total or the updated Lagrangian formulation should be used throughout for all element types.
2) If a structure is to be modeled by a mixture of different element types, it appears that the user can only combine

---

* Definitions of total and updated Lagrangian formulations are given in Section III.

the truss and 2-D solid elements. Any other combination of different element types will not satisfy the compatibility condition at the junction of the nodes due to the difference in degrees of freedom. In order to remove this difficulty, appropriate constrain conditions must be imposed at the nodes involved.

3. Material Models

In finite element analysis, the definition of constitutive relations of a material is needed for calculation of the stiffness matrices and stress components at the element level. The material models included in ADINA, as seen in Table 1, are element - dependent. That is to say, if the material model is available to one element type, it may not be available to other element types. The 2/D continuum element has the most complete list of material models ranging from linear to nonlinear stress-strain relationships. For a quick reference, the material models as opposed to the element library in ADINA are shown in Table 2. As seen from this table, the element that has the most material model coverage is the 2/D continuum.

In addition, an "element birth and death" option is available for all elements and material models in ADINA. The element birth option will activate an element, which is initially not active, at a specified time of birth. On the other hand, the death option makes an active element to become inactive at its time of death.

Table 2. Availability of Material Models vs. Element Library

Material Models	Truss	2/D Continuum	3/D Continuum	Beam
Isotropic linear elastic	x	x*	x*	x
Orthotropic linear elastic		x	x	
Isotropic thermo-elastic	x	x	x	
Nonlinear elastic	x			
Curve description model		x	x	
Concrete model		x	x	
Drucker-Prager model+		x		
von Mises elastic-plastic with isotropic or kinematic hardening	x	x	x	x
Thermo-elastic-plastic and creep with isotropic or kinematic hardening	x	x	x	
Mooney-Rivlin material		x†		

Notes:

* In the case of large deformation analysis, the isotropic linear elastic model can be used for both TL and UL formulations, whereas other material models are limited to TL formulation only.

\+ Drucker-Prager model can only be used for two-dimensional plane strain or axisymmetric deformation analysis

† Mooney-Rivlin material can only be used for plane stress analysis with the TL formulation.

According to the manual, these options are useful in the analysis of construction or excavation processes. It is not clear, however, to the evaluators what theoretical implication these options will cause. Furthermore, sudden declaration of element birth or death during the critical stage of loading may introduce numerical in - stability when iterations are being performed. Therefore, the user must apply the element birth or death option with some care.

4. Other Features

As seen in Table 1, one of the major improvements of ADINA over its predecessor NONSAP is the availability of the out-of-core linear equation solver. This option provides greater flexibility for the user to handle large size application problems on most of the main frame computers. Other useful improvements over the NONSAP include

1) Calculation of pressure loading
2) Thermal loading
3) A companion temperature analysis code ADINAT [9]

From the user's standpoint, the biggest drawback of ADINA is probably the lack of a comprehensive pre- and post-processing capability. The pre-processing in terms of automatic mesh generation and on-line display of finite element grid with editing capability is extremely useful for conducting nonlinear analysis of large scale structures. Nevertheless, the program provides the option to read the input information from a porthole tape on which the data can be generated from an independent preprocessor.

Data check option is available, but it is far from sufficient. The current option only prints out the input data. Some preliminary calculations, e.g. the area or volume of the element, determinant of the Jacobian matrix, etc., should be included to check any mistakes made in nodal coordinates and element-nodal number relationship. To further demonstrate this point, for example, the most frequently encountered error message in using ADINA is the one printed in the equation solver "COLSOL", i.e.

"STIFFNESS NOT POSITIVE DEFINITE
NONPOSITIVE PIVOT FOR EQUATION XX
PIVOT = (Value of diagonal term)"

The reason for causing this error can be many. Typically, they are

1) Mistakes in nodal coordinates
2) Mistakes in element - nodal number definitions
3) Erroneous material constants
4) Accumulated numerical errors due to insufficient iterations
5) The problem has reached structural or material instability.

Because of the above mentioned sources of errors which are difficult to pin-point, it is imperative for a nonlinear analysis to go through a more thorough data checking process.

5. Review of Documentations

ADINA documents consist of primarily the program's instruction manual [2] and the theoretical manual [3]. In the theoretical manual, a description of the theoretical basis, finite element discretization, definitions of material models, numerical approach and solution of sample problems are systematically presented. This manual provides a sound basis for the numerical algorithm undertaken in the code. The instructional manual consists of three main parts: i) Description of ADINA, ii) Sample problems, and iii) User's instruction. These documents have been reviewed and several comments are drawn.

Commentary:

1) Theoretical manual is well documented, in which the element stiffness matrices and definitions of various material models adopted in the code are available for users who wish to understand the coding procedure.

2) For large deformation analysis, both the total Lagrangian (TL) and the updated Lagrangian (UL) formulations are given for 2/D as well as 3/D continuum elements. At several places in the manual, it was stated that the choice between the TL and UL formulations lies essentially on their relative numerical efficiency. From the stiffness formulation of the 2/D element, for example, the linear part of the strain-displacement transformation matrix for the TL formulation is a full matrix, whereas for the UL formulation is sparse. Consequently, the UL formulation appears to be more numerically efficient than the TL formulation. Following this implication, the user may become pursuaded to use the UL formulation for all situations without considering other important factors such as the suitability of constitutive relations for a given material, etc.

3) The user's instruction manual was clearly written and fairly easy to follow. Even for a new user of the ADINA code, the input data can be prepared according to the instructions without much confusion.

4) For the data input some of the control variables used in the program could be consolidated so that the number of input blank cards can be reduced. For example if a control variable is provided on input Card 1 of Section II to distinguish the static, dynamic or frequency analysis option, then the blank cards #2, #3, and #4 can be eliminated when a static analysis is requested.

5) Although ADINA does not have a sophisticated pre- or post-processor package, the program has already set up the ground work to read the nodal and element data from a tape created by a pre-processor. Also, the analysis results, such as nodal displacements and element stresses, generated by ADINA can be written on an output tape for plotting or further evaluation by a post-processor. Therefore, the program can easily be interfaced with a pre- and post-processing software.

6) In an attempt to achieve better computational efficiency, the program provides the options to the users that stiffness reformation and equilibrium iterations can be requested for certain blocks of time steps. However, it is not clear how these options can be effectively utilized unless the analyst knows the convergence characteristics of a nonlinear problem a priori. The reviewers feel that the option on iteration time blocks may have

greater application in handling the elastic-plastic problems for which convergent solution is difficult to obtain when the structure is experiencing plastic loading and then unloading. This point will be further demonstrated in Part II of this report [10].

III. THEORETICAL BASIS

As mentioned earlier, the ADINA program can perform both linear and nonlinear, static and dynamic analysis for a range of structures. Both geometric and material nonlinearities were included in its theoretical formulations. In this section, the basic theory, constitutive relations, finite element idealization, and numerical scheme for both static and dynamic analysis will be reviewed and commented.

1. Theory

The solution approach adopted in ADINA is basically an incremental method for solving nonlinear problems, whether static or dynamic. For each loading increment, a tangent stiffness is used in the solution of displacement finite element equations and the solution is improved by invoking equilibrium iterations, while the stiffness is kept constant. This method corresponds to the so-called modified Newton-Raphson method. For the treatment of large deformation, both the total Lagrangian and updated Lagrangian formulations are adopted. A brief review on each of the two formulations is given as below.

The TL formulation for large deformation finite element analysis has been adopted by a number of researchers, and the works that can be cited are, for example, Martin [11], Oden [12], Hibbitt, Marcal and Rice [13], McNamara [14], Sharifi and Yates [15]. Needleman [16] and Hutchinson [17] also used the TL formulation with convected coordinates to study finite strain plasticity problems. For the purpose of discussion, a brief review of this formulation is given.

For a nonlinear finite element system, the incremental equations of equilibrium are generally derived from a virtual work principle. In reference to a total Lagrangian description, the equilibrium condition of a structure at time t+Δt can be stated by the principle of virtual displacement [2,18], i.e.

$$\int_{{}^{o}V} {}_{o}C_{ijrs}\; {}_{o}\varepsilon_{rs}\; \delta\, {}_{o}\varepsilon_{ij}\; {}^{o}dv + \int_{{}^{o}V} {}^{t}_{o}S_{ij}\; \delta_{o}\eta_{ij}\; {}^{o}dv$$

$$= {}^{t+\Delta t}R - \int_{{}^{o}V} {}^{t}_{o}S_{ij}\; \delta_{o}e_{ij}\; {}^{o}dv \qquad (1)$$

where $${}^{t+\Delta t}R = \int_{{}^{o}A} {}^{t+\Delta t}_{o}t_{K}\; \delta u_{K}\; {}^{o}da + \int_{{}^{o}V} {}^{o}\rho\; {}^{t+\Delta t}_{o}f_{K}\; \delta\, u_{K}\; {}^{o}dv \qquad (2)$$

and ^{o}A = Area of body in the configuration at time o.

$_{o}C_{ijrs}$ = Component of tangent constitutive tensor at time t referred to the configuration at time o.

$_{o}\varepsilon_{ij}$ = Incremental Green-Lagrange strain tensor referred to the configuration at time o

${}^{t}_{o}S_{ij}$ = 2[nd] Piola-Kirchhoff stress tensor in the configuration at time t

$_{o}e_{ij}$ = Linear part of strain increment $_{o}\varepsilon_{ij}$

$_{o}\eta_{ij}$ = Nonlinear part of strain increment $_{o}\varepsilon_{ij}$

${}^{t+\Delta t}_{o}t_{i}$ = Component of surface traction vector per unit area, at time t+Δt, referred to the configuration at time o

u_{i} = Component of incremental displacement

$^{o}\rho$ = Specific mass of the body in the configuration at time o

${}^{t+\Delta t}_{o}f_{K}$ = Component of body force vector per unit mass, at time t+Δt, referred to the configuration at time o.

In the case of dynamic analysis, a term due to inertia, i.e. $(-\int {}^{o}\rho\, {}^{t+\Delta t}\ddot{u}_{K}\, \delta u_{K}\, {}^{o}dv)$, must be added to the left hand side of Eq. (1). The stress and strain components at time t+Δt are calculated from the following relations:

$$ {}^{t+\Delta t}_{o}S_{ij} = {}^{t}_{o}S_{ij} + {}_{o}S_{ij} \qquad (3)$$

$$ {}^{t+\Delta t}_{\ \ o}\varepsilon_{ij} = {}^{t}_{o}\varepsilon_{ij} + {}_{o}\varepsilon_{ij} \tag{4} $$

and
$$ {}_{o}\varepsilon_{ij} = {}_{o}e_{ij} + {}_{o}\eta_{ij} \tag{5} $$

$$ {}_{o}e_{ij} = \frac{1}{2}\left({}_{o}u_{i,j} + {}_{o}u_{j,i} + {}^{t}_{o}u_{K,i}\ {}_{o}u_{K,j} + {}_{o}u_{K,i}\ {}^{t}_{o}u_{K,j}\right) \tag{6} $$

$$ {}_{o}\eta_{ij} = \frac{1}{2}\ {}_{o}u_{K,i}\ {}_{o}u_{K,j} \tag{7} $$

where
$$ {}_{o}u_{i,j} = \frac{\partial\,{}_{o}u_{i}}{\partial\,{}_{o}x_{j}} \tag{8} $$

$$ {}^{t}_{o}u_{i,j} = \frac{\partial\,{}^{t}_{o}u_{i}}{\partial\,{}_{o}x_{j}} \tag{9} $$

In order to make Eq. (1) solvable, the increment of strain components ${}_{o}\varepsilon_{ij}$ is approximated by

$$ {}_{o}\varepsilon_{ij} \simeq {}_{o}e_{ij} \tag{10} $$

Therefore, Eq. (1) becomes

$$ \int_{o} C_{ijrs}\ {}_{o}e_{rs}\ {}_{o}e_{ij}\ {}^{o}dv + \int {}^{t}_{o}S_{ij}\ \delta\ {}_{o}\eta_{ij}\ {}^{o}dv $$

$$ = {}^{t+\Delta t}R - \int_{{}^{o}V} {}^{t}_{o}S_{ij}\ \delta\ {}_{o}e_{ij}\ {}^{o}dv \tag{11} $$

This is the virtual work equation based on which the total Lagrangian finite element formulation was derived in ADINA.

It is seen that the major assumption made in arriving at Eq. (11) is the linearization of the incremented strain tensor by replacing ${}_{o}\varepsilon_{ij}$ by ${}_{o}e_{ij}$ in the first term on the left hand side of Eq. (1). The implication of this approximation is that the strain increment in an element of material for each time (or loading) step must be restricted to be small. Otherwise, the equilibrium condition of the structure cannot be satisfied. Another approximation made is the evaluation of virtual external work on the right hand side of Eq. (1). Generally, in the case of large deformation the virtual external work should consist of two parts: a part due to the change of load value itself, and another part due to the change in geometry while the load value remains constant [13, 19, 20]. In the ADINA program, the latter part, i.e., the effect of deformation dependent load, was neglected.

The updated Lagrangian formulation uses the current configuration of a body, i.e. at time t, under consideration as a reference state to obtain the state of deformation at next time step $t+\Delta t$. This approach has previously been used by Yaghmai and Popov [21], Sharifi and Popov [22], Hofmeister, Greenbaum and Evenson [23]. Osias and Swedlow [24] applied the UL formulation for the solution of large elastic-plastic problems with a Galerkin procedure which admits non-symmetric constitutive relations. Based on Hill's work [25], McMeeking and Rice [26], Yamada [27], Yamada, Hirahawa and Wifi [20], and Namat-Nasser and Taya [28] have also presented a UL formulation in dealing with large strain plasticity problems. An improved UL formulation to deal with the numerical difficulty arose due to the incompressible behavior in large strain plasticity was presented by Atluri [29], Murakawa and Atluri [30].

According to ADINA, the virtual work principle corresponding to the UL formulation is given by

$$\int_{{}^tV} {}_tC_{ijrs}\ {}_te_{rs}\ {}_te_{ij}\ {}^tdv + \int_{{}^tV} {}^t\tau_{ij}\ \delta_t\eta_{ij}\ {}^tdv$$

$$= {}^{t+\Delta t}R - \int_{{}^tV} {}^t\tau_{ij}\ \delta_t e_{ij}\ dV \tag{12}$$

where ${}_tC_{ijrs}$ = Component of constitutive tensor at time t referred to the configuration at time t.

${}_t\varepsilon_{ij}$ = Incremental Green-Lagrangian strain tensor referred to the congifiguration at time t

${}_te_{ij}$ = Linear part of ${}_t\varepsilon_{ij}$

${}_t\eta_{ij}$ = Nonlinear part of ${}_t\varepsilon_{ij}$

${}^t\tau_{ij}$ = Cauchy stress tensor in the configuration at time t.

${}^tV$ = Volume of the body occupied at time t.

The stress and strain components at time t+Δt are calculated from

$${}^{t+\Delta t}_{\ \ t}S_{ij} = {}^t\tau_{ij} + {}_tS_{ij} \tag{13}$$

$${}^{t+\Delta t}_{\ \ t}\varepsilon_{ij} = {}_t\varepsilon_{ij} = {}_te_{ij} + {}_t\eta_{ij} \tag{14}$$

$${}_te_{ij} = \frac{1}{2}\left({}_tu_{i,j} + {}_tu_{j,i}\right) \tag{15}$$

$${}_t\eta_{ij} = \frac{1}{2}\ {}_tu_{K,i}\ \ {}_tu_{K,j} \tag{16}$$

with
$${}_tu_{i,j} = \frac{\partial u_i}{\partial\, {}^tx_j} \tag{17}$$

Therefore, Eq. (12) forms the basis for the UL finite element formulation in ADINA. Similar to the TL formulation, this equation involves two approximations: i) the strain increment ${}_t\varepsilon_{ij}$ in the first term on the left hand side of Eq. (12) was linearized to become ${}_te_{ij}$, and ii) the virtual variation of external work due to any geometric change was neglected.

It is also important to note that for large deformation analysis appropriate definition for the constitutive tensor ${}_tC_{ijrs}$ must be assigned to relate the stress ${}_tS_{ij}$ and strain ${}_t\varepsilon_{ij}$, and this point will be further discussed in the next section.

Although both the TL and UL formulations were presented in the ADINA theoretical manual [2], they are not available for all the elements. As seen in Table 3, only the 2/D and 3/D continuum elements were derived for both formulations.

Table 3. Large Deformation Formulations

Element Type	Total Lagrangian	Updated Lagrangian
Truss Element		X*
2/D Continuum	X	X+
3/D Continuum	X	X+
Beam Element		X*
Shell Element†	X	
2/D Fluid†		X
3/D Fluid†		X

Notes:

* . Large displacements but small strains.
+ . Only isotropic linear elastic material model can be used.
† . Element is available in 1978 version.

2. Constitutive Relations

One of the important features of a finite element program is its ability to model the responses of various materials. In this connection, ADINA has included a number of material models ranging from linear to nonlinear behavior. With the consideration of the kinematics in its general form as seen in the virtual work equations, i.e. Eqs. (11) and (12), it appears that the program can be applied to any type of nonlinear analysis of structural problems involving either nonlinear material, large deformation or both. However, care must be given when both material and geometric nonlinearities are to be considered simultaneously. In fact, the version of ADINA being reviewed is only suitable for large deformation analysis in the sense of large displacement but small strain (except for the Mooney-Rivlin material model). The discussion on this point can be made by reviewing some of the fundamentals in constitutive theory as follows.

The constitutive relations of a material can be expressed in different forms in conjunction with the stress and strain measures chosen. Following ADINA's notations, three different expressions are to be discussed: Incremental Piola-Kirchhof stress vs. incremental Green strain, updated incremental Piola-Kirchhof stress vs. updated incremental Green strain, and Jaumann rate of Kirchoff stress vs. deformation rate. These relationships are given by

$$ {}_0S_{ij} = {}_0C_{ijrs} \; {}_0\varepsilon_{rs} \tag{18} $$

$$ {}_tS_{ij} = {}_tC_{ijrs} \; {}_t\varepsilon_{rs} \tag{19} $$

$$ {}_{t}S^{\nabla}_{ij} = {}_{t}C^{\nabla}_{ijrs} \quad {}_{t}d_{rs} \tag{20}$$

where ${}_{t}S^{\nabla}_{ij}$ = Jaumann (frame invariant) rate of Kirchhoff stress tensor in the configuration at time t.

${}_{t}C^{\nabla}_{ijrs}$ = Constitutive tensor relating ${}_{t}S^{\nabla}_{ij}$ and ${}_{t}d_{rs}$ in the configuration at time t

${}_{t}d_{rs}$ = Deformation rate tensor

$$ = \tfrac{1}{2}({}_{t}u_{r,s} + {}_{t}u_{s,r}) \tag{21}$$

In the above equations, the consitutive tensors are related through

$$ {}_{o}C_{ijrs} = \frac{{}^{o}\rho}{{}^{t}\rho} \; {}^{o}_{t}x_{i,m} \; {}^{o}_{t}x_{j,n} \; {}^{o}_{t}x_{r,p} \; {}^{o}_{t}x_{s,q} \; {}_{t}C_{mnpq} \tag{22}$$

and

$$ {}_{t}C_{ijrs} = {}_{t}C^{\nabla}_{ijrs} - {}^{t}\tau_{ir}\,\delta_{js} - {}^{t}\tau_{jr}\,\delta_{is} + {}^{t}\tau_{ij}\,\delta_{rs} \tag{23}$$

where ${}^{t}\rho$ = Specific mass of the body in the configuration at time t

${}^{o}_{t}x_{i,m}$ = Derivative of coordinate in the configuration at time o with respect to the coordinate at time t

$$ = \frac{\partial}{\partial\, {}^{t}x_{m}} ({}^{o}x_{i}) $$

${}^{t}\tau_{ij}$ = Cauchy stress tensor at time t.

The choise on the use of Eq. (18), (19) or (20) depends on the type of material in question. For example if a material is hypereleastic for which the constitute relations are derivable from a strain energy function [26], i.e.,

$$ {}^{t+\Delta t}_{\ o}S_{ij} = \frac{\partial W}{\partial\, {}^{t+\Delta t}_{\ o}\varepsilon_{ij}} \tag{24} $$

then Eq. (18) would be the natural choice for the description of material response. On the other hand, for hypoelastic materials such as metal plasticity, the constitutive behavior depends on the stress and strain history developed in the structure and the natural description of the material response would be to relate the frame invariant stress rate and the deformation rate as given in Eq. (20). As discussed by Hibbitt, Marcal and Rice [13] in accordance with Hill's interpretation [27], Eq. (20) reduces to the true stress vs. logarithmic strain relation in simple tension of metals.

From the theoretical standpoint, it is indifferent, in the finite element derivation, to choose one form of constitutive description over the other, as long as the transformation relations, i.e. Eq. (22) or (23) are properly defined in the program. However, from the numerical point of view, it is preferable to use the stress and strain measures in the finite element computations same as those for the constitutive description. Although the transformations between the constitutive tensor can be made according to Eq. (22) or (23), such calculations are very time consuming and this point was previously discussed in references [13, 27]. Therefore, for large strain analysis of elastic materials, it is more efficient to use the TL approach; whereas for large strain elastic-plastic analysis, the UL approach would be the natural choice.

As we turn our attention to the ADINA program, most of the nonlinear material models are restricted to the TL formulation and type of transformation between the constitutive tensors as given in Eqs. (22) and (23) is not available in the code. It is therefore concluded that the present version of ADINA is only suitable for small strain, nonlinear material analysis or, at most, for large displacement nonlinear material analysis.

The second point worth noting is the precise physical nature of the incremental stresses ${}_{o}S_{ij}$ and ${}_{t}S_{ij}$. Since ${}_{o}S_{ij}$ represents the incremental 2nd Piola-Kirchhoff stress tensor at time t referred to the configuration at time o, it is a convected quantity deforming together with the material element of a body. The stress quantity ${}_{t}S_{ij}$ has similar definition as ${}_{o}S_{ij}$ except that it is referred to the updated configuration at time t. That is to say, ${}_{t}S_{ij}$ is not deforming with the material fibers. For an orthotropic or anisotropic material undergoing large displacements, the use of ${}_{t}S_{ij}$ in the constitutive description will not be appropriate unless the effect of material axes rotations is accounted for. Such option is not available in ADINA. Nevertheless, the program does impose the necessary restriction that only the TL formulation is permitted for the case of large deformation analysis of orthotropic materials.

Another important note is the limitation of the program in dealing with material incompressibility. In engineering application, the elastic material most frequently used for large strain analysis is the Mooney-Rivlin (or rubber-like) material. In order to handle the general case nonlinear analysis of such material, an incompressible finite element formulation [7] is necessary. Moreover, in large strain elastic-plastic analysis such as metal-forming problems, plastic incompressibility condition introduces two numerical difficulties in the finite element analysis:
i) the element becomes overly constrained causing unnecessary stiffening effect [31] , and ii) small time (or loading) increment is necessary

in order to obtain convergent solution [29] . From this discussion, it appears that ADINA will have greater versatility if a finite element formulation with incompressibility constraint is included in the future extension of the code.

As mentioned earlier, the material models in ADINA are element-dependent, both the 2/D and 3/D continuum elements have a fairly extensive list of material models. From the theoretical consideration, the material models included in ADINA can be classified in two categories: i) classical models, which conform to the basic principles in mechanics, and ii) empirical models, which are largely based on experimental observations and/or conjectures. In accordance with these classifications, the material models are listed as the following:

<u>Classical Models</u>

Isotropic linear elastic

Orthotropic linear elastic

Isotropic thermo-elastic

Mooney-Rivlin material

Drucker-Prager model (Elastic-perfectly plastic material)

von Mises elastic-plastic with isotropic or
kinematic hardening

Thermo-elastic-plastic and creep

<u>Empirical Models</u>

Curved description model

Concrete model

In addition to the above discussion, some further comments can be made:

1) The von Mises elastic-plastic model with either isotropic or kinematic hardening is restricted to materials with bilinear stress-strain curve in uniaxial response. This limitation is unnecessary for the isotropic hardening materials and the code could be modified to include the nonlinear stress-strain response.

2) The curve description model was intended for simulating the nonlinear response of geological materials [32]. In formulating the constitutive relations, a number of assumptions were made. First, the incremental stresses and strains are related by instantaneous moduli in shear and bulk, respectively.

$$S_{ij} = 2\,{}^{t}G\, g_{ij} \tag{25}$$

$$\sigma_m = 3\,{}^{t}K\, e_m \tag{26}$$

where S_{ij} = Incremental deviatoric stresses

g_{ij} = Incremental deviatoric strains

σ_m = Incremental mean stress

e_m = Incremental mean strain

${}^{t}K$ = Instantaneous bulk modulus, which is a function of loading condition and volume strain. One set of input data is required for loading and another set for unloading.

${}^{t}G$ = Instantaneous shear modulus, which is also a function of loading condition and volume strain. Only one set of input data is necessary for loading condition. The values for unloading are scaled by the stress-strain data in bulk.

The loading and unloading conditions of the material are specified from a strain criterion: If the calculated ${}^{t}e_m \leq e_{min}$ (a minimum mean strain), the material is under loading, and if ${}^{t}e_m > e_{min}$, the material is under unloading. In addition, the material may weaken or crack when the maximum principal stresses exceed certain limit.

It is important for the user to note three possible limitations of this model:

i) The material behavior is uniquely determined by the magnitude of volume strain without regard to its deformation history although different stress-strain curves are used for loading and unloading.

ii) The constitutive relationship assumes that the principal axes of the stress increment coincide with those of the strain increment. This is generally not the case due to material - induced anisotropy.

iii) The relationships in Eqs. (25) and (26) are equivalent to a non-associated flow rule which is known for lack of solution uniqueness, excepted in a very specialized case [33].

3) The concrete model in ADINA, although involves some sophisticated assumptions [34], is very similar to the curve description model. Therefore, this model is also subject to the limitations cited above. Furthermore, to model the nonlinear-fracturing behavior of concrete materials, a number of different constitutive theories have been proposed in the literature [35-39] and the user should make proper judgement on the selection of the models that are available.

3. Finite Element Idealization

In ADINA, the truss, 2/D continuum and 3/D continuum elements are the isoparametric family except the beam element for which independent interpolation functions are used to approximate the displacement field. The use of isoparametric elements has several apparent advantages [7,40]:

i) The elements satisfy the convergence requirements of finite elements. That is, rigid body modes, constant strain behavior and inter-element compatibility are preserved.

ii) Various structural configurations and curved boundaries can be modeled.

iii) Different material models can easily be incorporated into the stiffness matrix by numerical integration.

iv) The elements give better stress field as compared to the triangular or tetrahedron elements.

For a quick reference, the element types vs. degrees of freedom and order of numerical integration are shown in Table 4. A brief outline on the finite element approximations for each element type is given below.

3.1 Truss Element

It represents a structural member capable of transmitting the axial force only. The element may have 2, 3, or 4 nodes, and each node has 3-degrees of freedom in translation (Fig. 1). The user may also have the option to use the truss element for axisymmetric analysis in which case only one node is defined in the YZ-plane. The inter-

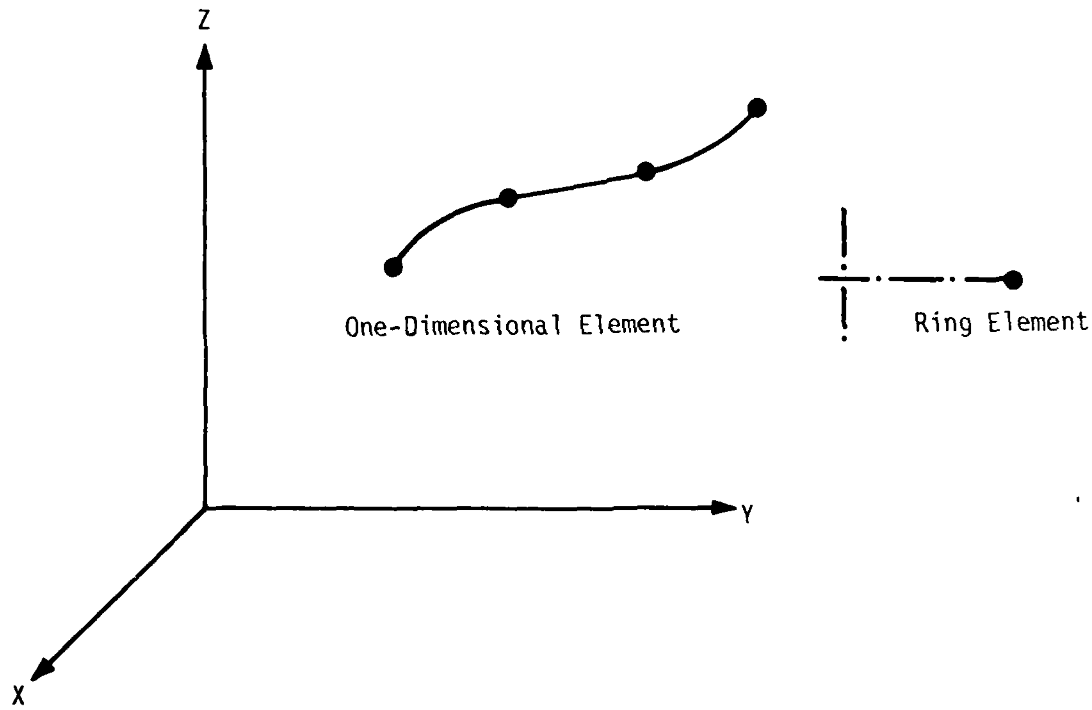


Fig. 1. Truss Element

Table 4. Order of Numerical Integration

Element Type	No. of Nodes per Element	DOF per Node	Order of Numerical Integration
Truss	2 - 4	3	2 - 4
2/D Continuum	4 - 8	2	2 - 4
3/D Continuum	8 - 21	3	2 - 4
Beam	2	6	r-direction: 1-7 s-direction: 1-7 t-direction: 1-8

polation functions used for the element formulation can be found from references [41,42]. Since the cross sectional area was assumed to remain constant during deformation, the element can be applied only to large displacement analysis with small strains. Large deformation analysis is based on the UL formulation.

3.2 2/D Continuum Element

It is an isoparametric element with 4 - 8 nodes, and 2-degrees of freedom at each node as seen in Fig. 2. The interpolation functions of this element can be found from references [7,41]. Although the user has the option to choose any number of nodes between 4 and 8 to describe an element, use of 5- node and 7- node elements should be avoided due to the incompatible displacement field in the element which may cause numerical difficulty [43].

The element stiffness, consistent mass and internal force are evaluated by numerical integration with Gauss quadrature formulae. The user must specify the order of numerical integration varying from 2 to 4. Generally for linear elastic material, an integration order of 2 is sufficient. However, for nonlinear material in which high stress gradients occur, an integration order of 3 is recommended. The program sets the integration order of 3 for the evaluation of consistent mass matrix.

A triangular element can be obtained by degenerating a 4- node or 8- node quadrilateral element as shown in Fig.3. The use of degenerated 8-node triangular element should be avoided since the element stiff-

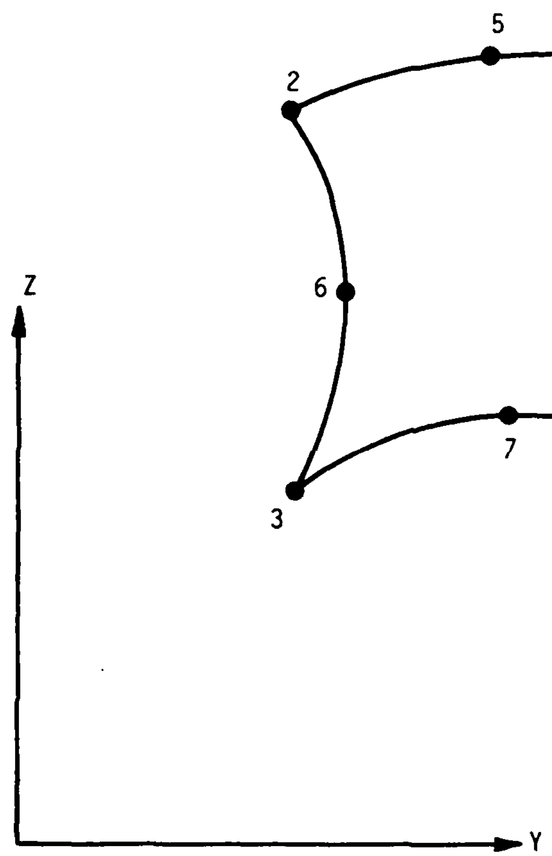


Fig. 2. 2/D Continuum Element

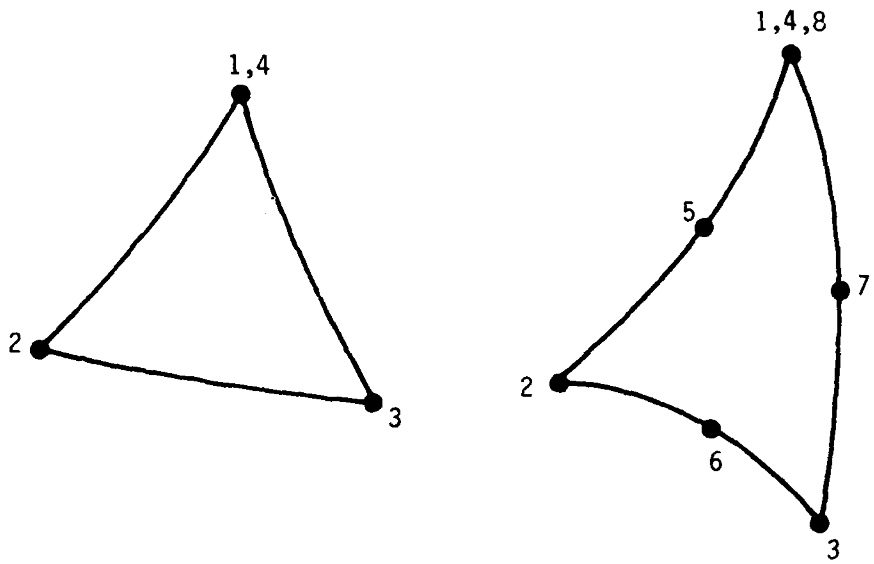


Fig. 3. Degenerated Triangular Elements

ness is biased depending on the manner that the local nodes are combined [7,40]. However, the degenerated 8- node element with quarter point has proven to be useful to model the crack tip singularity in fracture mechanics [44].

3.3 3/D Continuum Element

The numerical approximation of this element [7,41] is basically the same as the 2/D element, therefore, most of the discussion given in the above is also applicable here. The element consists of 8 - 20 nodes with 3- degrees of freedom at each node (Fig. 4). Gauss quadriture formulae is used in numerical integration with the integration order varying between 2 and 4. The user may specify the integration order in the surface and over the thickness of the element independently. Since the computational cost for evaluating the stiffness matrix of a 3/D element is rather high and it is closely associated with the number of integration points chosen, the user must pay extra care in making such a decision. In most cases, an integration order of 2 is sufficiently accurate. However, in case of high stress gradients or nonlinear materials, an integration order of 3 is necessary. If this element is to be applied to the bending analysis of a thin plate or a thin shell, reduced order of numerical integration must be used in the plane of the element in order to obtain any meaningful results. This is further discussed in Part II in the report.

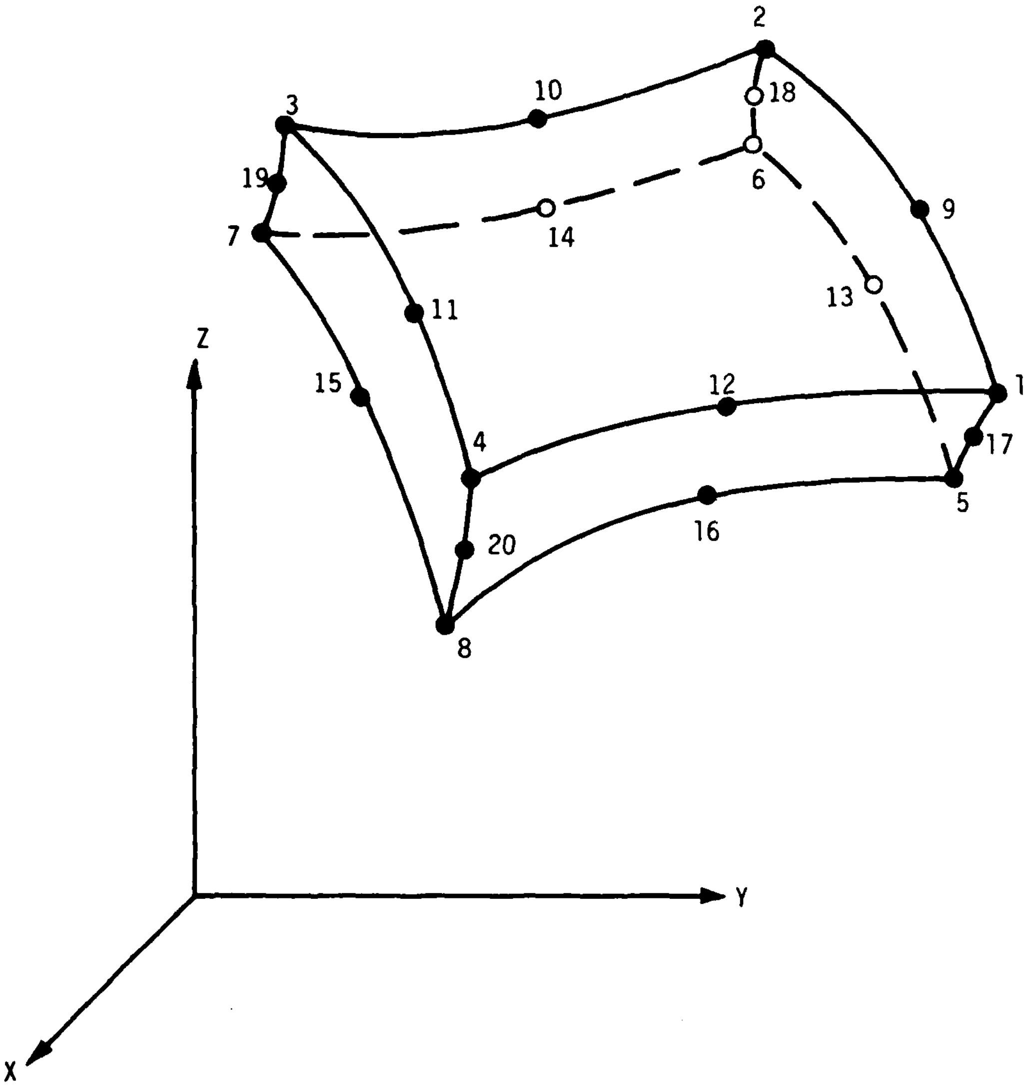


Fig. 4. 3/D Continuum Element

3.4 Beam Element

Differing from the previous three elements, the beam element is not a member of the isoparametric family. The element is described by two nodes at the ends with 6- degrees of freedom at each node: 3- translations and 3- rotations (Fig. 5). In the finite element formulation, the following assumptions were made:

i) Only prismatic straight beam was considered.

ii) Plane sections of the beam remain plane during deformation, but not necessarily perpendicular to the neutral axis, i.e., a constant shear is allowed.

iii) The beam undergoes large rotations but small strains.

iv) A cubic variation in bending displacement and linear variation in axial and torsional deformations.

v) The UL formulation was adopted.

The interpolation functions for the displacement field are based on those in reference [42] and the derivation of nonlinear stiffness matrices are given in [45]. Numerical integration with Newton-Cotes formulas is used for the evaluation of nonlinear element stiffness matrix and internal force vector. The integration order varies from 5 to 7 in the longitudinal direction of the beam and 3 to 7 in the transverse directions. However, for a linear elastic beam under small deformation, the stiffness matrix is evaluated by close form formulations [42,46].

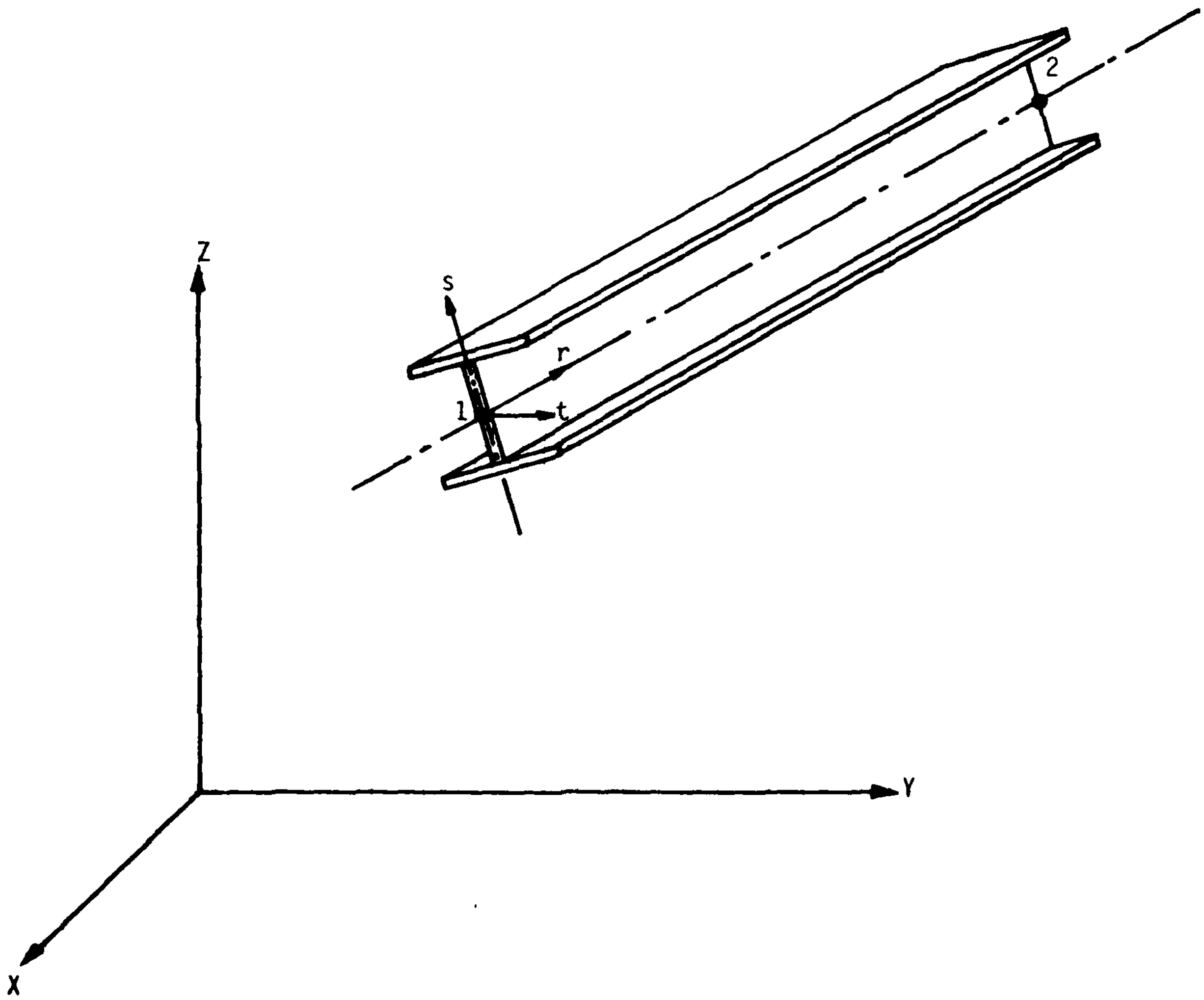


Fig. 5. Beam Element

4. Static Solution Method

The solution method employed in ADINA for solving nonlinear static problems is based on the incremental loading with tangent stiffness approach and equilibrium iterations, which corresponds to the Modified Newton-Raphson (MNR) procedure[7,23,47-49]. The user has the option to request the stiffness reformation at every loading increment or at specified loading increments. The user also controls the number of equilibrium iterations for convergence. If this number is not specified, the program set it to be "15" internally. The apparent advantage of this method is that it is computationally efficient for solving many practical nonlinear problems. Fig. 6 shows the convergence characteristics of the MNR method in the load-deflection plot of a structure exhibiting softening behavior. It is seen that if the load increment is sufficiently small, solution will converge monotonically. Convergence of the solution is determined in the program by checking the calculated displacements during iterations, i.e.

$$\frac{||\Delta U^{(i)}||}{||U^{(t+i-1)}||} \leq \text{RTOL} \qquad (27)$$

where $||\ ||$ denotes the Euclidean norm and

$\Delta U^{(i)}$ = Incremental displacement calculated up to the i-th iteration cycle.

$U^{(t+i-1)}$ = Total displacement calculated up to and including the current iteration cycle.

RTOL = A convergence tolerance specified by input, usually in the order of 0.01 or smaller.

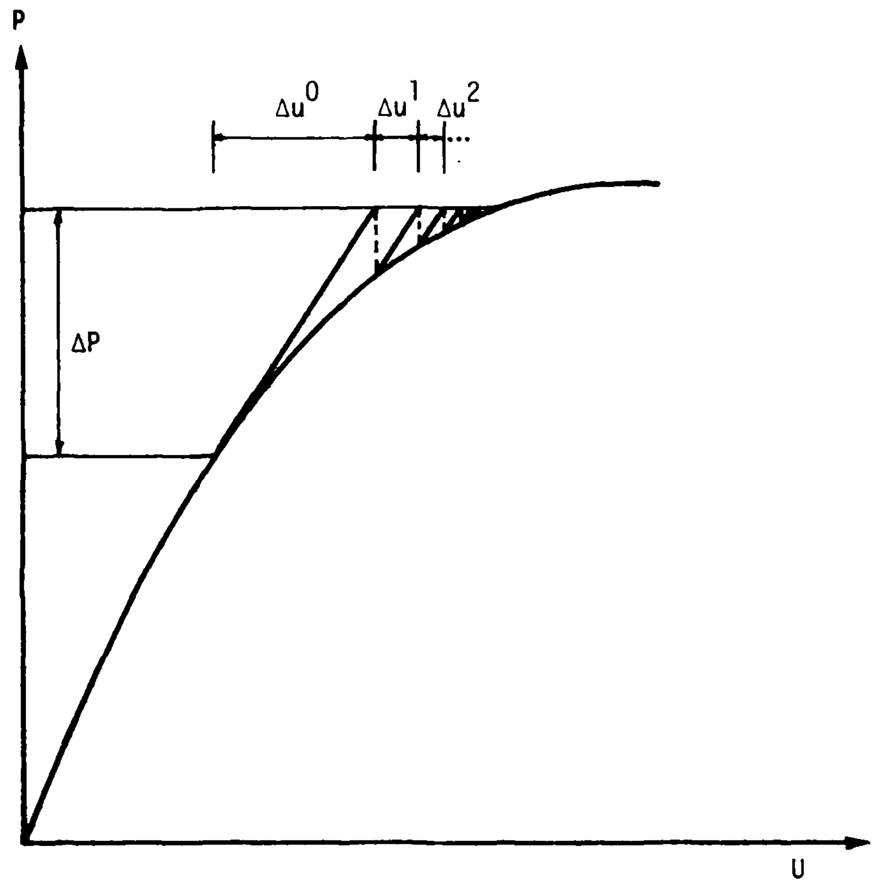


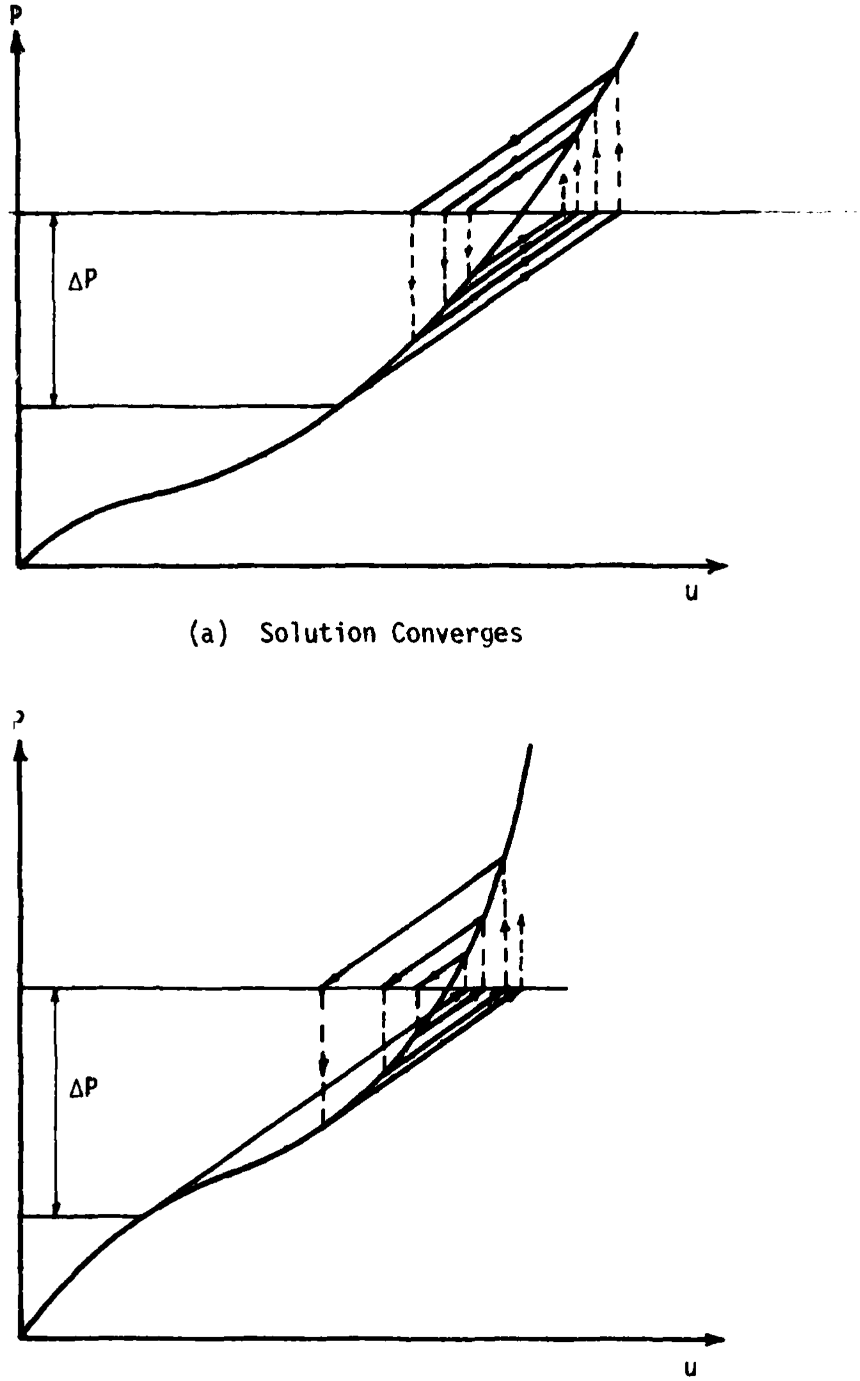
Fig. 6. Modified Newton-Raphson Procedure

In addition, the solution from iterations is declared to diverge if the norm of the out-of-balance load is larger than the norm of the incremental load for the current solution step.

Generally the MNR method does not give convergent solution under three conditions:

1) Load increment is too large.
2) Problems involve elastic-plastic deformations when elastic unloading occurs. One way to circumvene this problem is to either use elastic material law for the elements undergoing unloading or specify a very small load increment and suspend the equilibrium iteration just to overcome the elastic unloading.
3) The structure is becoming stiffened during deformation. In this case, the solution may or may not converge depending on the load increment specified or the load-deformation characteristics as seen in Fig. 7-a and 7-b. Even if the convergence does occur, the iterative displacements and residual forces are oscillatory and the convergence rate is generally slow.

Further discussion on the numerical characteristics of the MNR method will be given in Part II of this report.



(a) Solution Converges

(b) No Convergence

Fig. 7. Convergence Characteristics of a Stiffening Structure

5. Dynamic Analysis Method

With the inclusion of the inertia effect in the virtual work equation, i.e. Eq. (11) or (12), and after the finite element discretization, the equations of motion of a structure corresponding to either TL formulation or the UL formulation can be written in a typical matrix form at $t + \tau$

$$M^{t+\tau}\ddot{U} + C^{t+\tau}\dot{U} + K^{t+\tau}U = {}^{t+\tau}R - {}^{t}F \tag{28}$$

where

M = Mass matrix

C = Damping matrix

K = Structural stiffness matrix

U = Incremental nodal displacement vector

$\ddot{U}$ = Incremental nodal velocity vector

$\dot{U}$ = Incremental nodal acceleration vector

${}^{t+\tau}R$ = Total external nodal force vector at time $t+\tau$

${}^{t}F$ = Total internal nodal force vector at time t.

For nonlinear problems, the matrix K represents the tangent stiffness matrix of the finite element idealization which may include the effects due to geometric and material nonlinearities. The mass and damping matrices in the above equations are the same as those of a linear structure.

In ADINA, Eq. (28) is solved by using the direct integration method and both the implicit and explicit integration schemes with the following operators are available in the program:

Implicit Integration

i) Newmark β - method [50]

ii) Wilson θ - method [51]

Explicit Integration

i) Central - difference method [52].

The stability and accuracy of the aforementioned operators have been investigated quite extensively for linear structural dynamics by a number of investigators [53-60]. These studies showed that both the Wilson θ-method and Newmark β - method ($\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$) are unconditionally stable, whereas the central difference method is only conditionally stable. The stability condition for the central difference method is that the time step Δt must be smaller than a critical value Δt_{cr} [41] i.e.,

$$\Delta t \leq \Delta t_{cr} = \frac{T_n}{\pi}$$

where

T_n = smallest vibrational period of the finite element system to be considered.

From the study made by Goudreau and Taylor [57], the Wilson method was found to introduce considerable damping to the structure and the Newmark average acceleration scheme does not. They concluded that the use of Newmark method requires less than half as many time steps as the Wilson method.

While most experience on the stability and solution accuracy of the approximate operators was gained for the linear dynamic problems, little is known on the application of these operators to nonlinear problems. Weeks [61] has evaluated the Houbolt and Newmark operators for a one-degree-of-freedom geometrically nonlinear problem and found that both operators gave a satisfactory solution when the full Newton-Raphson iteration scheme is used. This result was also substantiated by McNamara [62] from his numerical investigations. From the energy considerations, the Newmark method with $\beta \geq \frac{1}{4}$ was also found to be unconditionally stable when applied to nonlinear structural dynamics [63-65]. Additional computational aspects in regard to the application of the direct integration method are discussed in [66-68].

Despite all the progress that has been made in structural dynamics, application of ADINA or any other computer codes to the analysis of nonlinear sturctural problems is a difficult task. The analysts are always confronted with two major issues:

i) A time step which can give convergent and reliable solutions

ii) Computational costs

Unfortunately, there are, as yet, no simple answers to these issues. From the literature, some of the points which might be helpful to the analysts are summarized as the following:

1) The apparent advantage of the explicit integration scheme over the implicit scheme is that for a lumped mass system it does not require the assemblage of structural and mass matrices, and matrix inversion which represent a major computational cost in the finite element analysis. However, this aspect is hampered by the fact that the explicit scheme is only conditionally stable in numerical integration and hence it requires smaller time steps as compared to the use of implicit schemes.

2) Some general discussions on the choice of the explicit and implicit schemes were given, for example, in [54, 56, 66, 68]. For wave propagation problems where high frequency response of the structure is more important and the discontinuity in velocity or acceleration persists, the explicit integration scheme will be the obvious choice. On the other hand, for structural dynamic problems where the response is predominated by the fundamental frequencies of the system, the implicit scheme is preferred.

3) When comparing the Newmark β-method vs. Wilson θ-method, in most cases both methods give the same results and have the same computational characteristics [41]. However, the Newmark method conserves the total energy for an undamped system whereas the Wilson method does not [57]. Therefore, the Newmark method seems to be superior. Furthermore, this method was proven to be unconditionally stable when applied to nonlinear dynamic problems.

4) For implicit integration, the step Δt should be smaller than $(1/20)T_n$, where T_n is the maximum period of the nonlinear system [69].

5) Computationally, it is more efficient to invoke equilibrium iterations in every loading step while the stiffness is updated at every 2-3 loading step.

IV. PROGRAMMING DESCRIPTION

In its early inception, a general purpose finite element structural software mostly represents an extension of special purpose finite element codes by utilizing certain common features:

1) Reading and processing of input data
2) Assemblage of the master matrix
3) Solution of large systems of matrix equations
4) Performing certain analysis procedures (i.e. static or dynamic, linear or nonlinear analysis)
5) Printing of output, etc.

With these features included, various element types and material descriptions can be added into the software to form a general purpose code for conducting a wide range of structural analysis. However, a large scale program without proper coding organization will soon become obsolete as it is expanded significantly beyond its original capability. More specifically, a poorly designed general purpose software may face the following potential problems:

1) It takes much longer computer time due to many conditional statements and unnecessary data transfers between the fast core and low speed storages.
2) It requires excessive computer storage for poor data management.
3) Coding changes become difficult without interfering with other parts of the software.

To alleviate some of these problems, a large scale general purpose software must be written on a modular basis and only a few modules are involved for a particular analysis. Moreover, all modules should be organized in a manner that they can be modified independently or changed without interfering with each other. In view of these considerations, the modern design of general purpose finite element software has evolved into a characteristic form. That is, the entire software may be divided into three major parts:

1) Input phase (pre-processor)
2) Analysis phase (main program)
3) Output phase (post-processor)

Since in ADINA, the output phase is a part of the analysis phase in terms of its coding organization, both items 2 and 3 will be reviewed as one unit. To provide a basis for discussion and to assist the ADINA user to follow its macro command, a flow diagram for the main module "AAMAIN" is included in the Appendix A1.

Also included in this chapter is a discussion on the use of control variables and data structure adopted in the code.

1. Input Phase

The major function of this phase is to read and/or generate input data which are necessary to define a structural problem. These include:

1) Read the definitions of master control variables, time history and integration method, printing options, etc.

Table 5. Input Sequence

Sequence	Subroutines	Flag	Input Description
1	ADINI	IND = 0	Definition of master control variables for solution options, time-history, and nodal print data.
2	ELCAL	IND = 0	Material data and element definitions.
3	ADINI	IND = 2	Applied loads: Concentrated forces or pressure load.

2) Read nodal constraint condition and coordinates, then generate the equation number matrix (ID-array) and store it on the low speed device, unit #8.

3) Read the initial conditions and initialize the nodal displacements, velocities and accelerations. Write this information on unit #8.

4) Read and generate element data: material properties and element-nodal number relations. From these, form the element connection arrays (LM) and store on: unit #1 - for linear elements, unit #2 - for nonlinear elements.

The above executions are done in three stages by calling the subroutines ADINI and ELCAL. The calling sequence is controlled in the main driver AAMAIN (as shown in A1) by use of a control variable IND (=0 or 2). This interrelationship is shown in Table 5. For further references, the subprograms called by ADINI and ELCAL are shown in sections A2 and A3 in the Appendix A, respectively.

It is noted that all input data must be provided according to the fixed format strictly following the sequential order in the user's manual. Reading and calculation of load data are made after the assemblage of linear structural stiffness (see flow diagram A1). This calling sequence cannot be easily reversed due to the dynamic allocations of array storage for the load calculations coded in the subroutine ADINI.

2. Analysis Phase

The analysis phase represents the main package of the Program which carries out the actual problem solving. The composition of this phase is generally made up by three libraries:

1) Procedure library - which consists of the solution options available to the user.
2) Geometric library - which includes various element types for the geometric modeling of a structure.
3) Material library - which contains various constitutive models for simulating the material behavior.

Each of the above categories is reviewed as below.

2.1 Procedure Library

The analysis procedures are primarily controlled by the main driver "AAMAIN". The options available include:

1) Linear static analysis
2) Linear dynamic analysis (direct step-by-step integration)
3) Nonlinear static analysis
4) Nonlinear dynamic analysis
5) Frequency analysis
6) For dynamic analysis, the choice of integration operators: Wilson θ, Newmark or central difference method.
7) In core or out-of-core solution (determined internally).
8) Reformation of structural stiffness
9) Iteration option for nonlinear static or dynamic analysis

Decision on the analysis options is made in AAMAIN by checking the pre-assigned values of the control variables (or flags) and subsequently, various analysis modules are called to carry out the analysis options requested by the user. An overview of the analysis procedures is shown in Fig. 8 and a detailed flow diagram for AAMAIN is given in Section A1 of the Appendix A.

As seen from Fig. 8 and Section A1, the modules corresponding to various analysis procedures were arranged in a serial form in AAMAIN. For any desired analysis option, a specific path is channeled through by combining appropriate modules or subroutines with the use of conditional statements. For discussion, the macro command of the analysis procedures (1) - (4) are listed in Tables 6-9 respectively. The calling sequences of the modules for different analysis procedures are quite similar with some minor variations. For example, to compare the algorithm difference between the linear static analysis and linear dynamic analysis, the macro command is almost identical except that three modules perform somewhat different calculations as seen in the following:

Module	*Linear Static Analysis*	*Linear Dynamic Analysis*
ASSEM	Form global stiffness	Form effective global stiffness, mass and damping matrices.
LOADEF	No calculations are performed	Form effective load vectors due to dynamics
NEWDAV	-	Update nodal displacements, velocities and accelerations

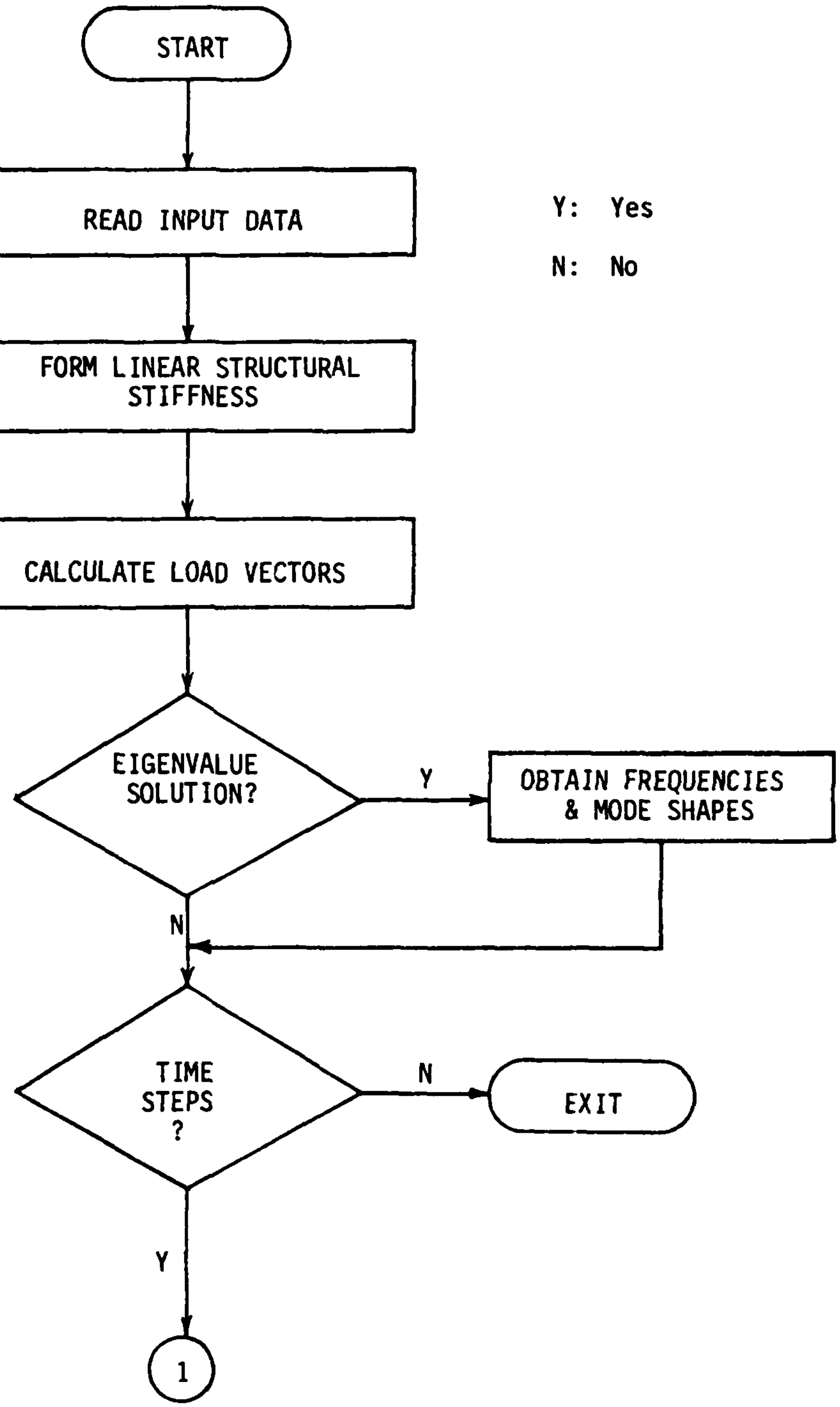


Fig. 8. An Overview of Analysis Procedures

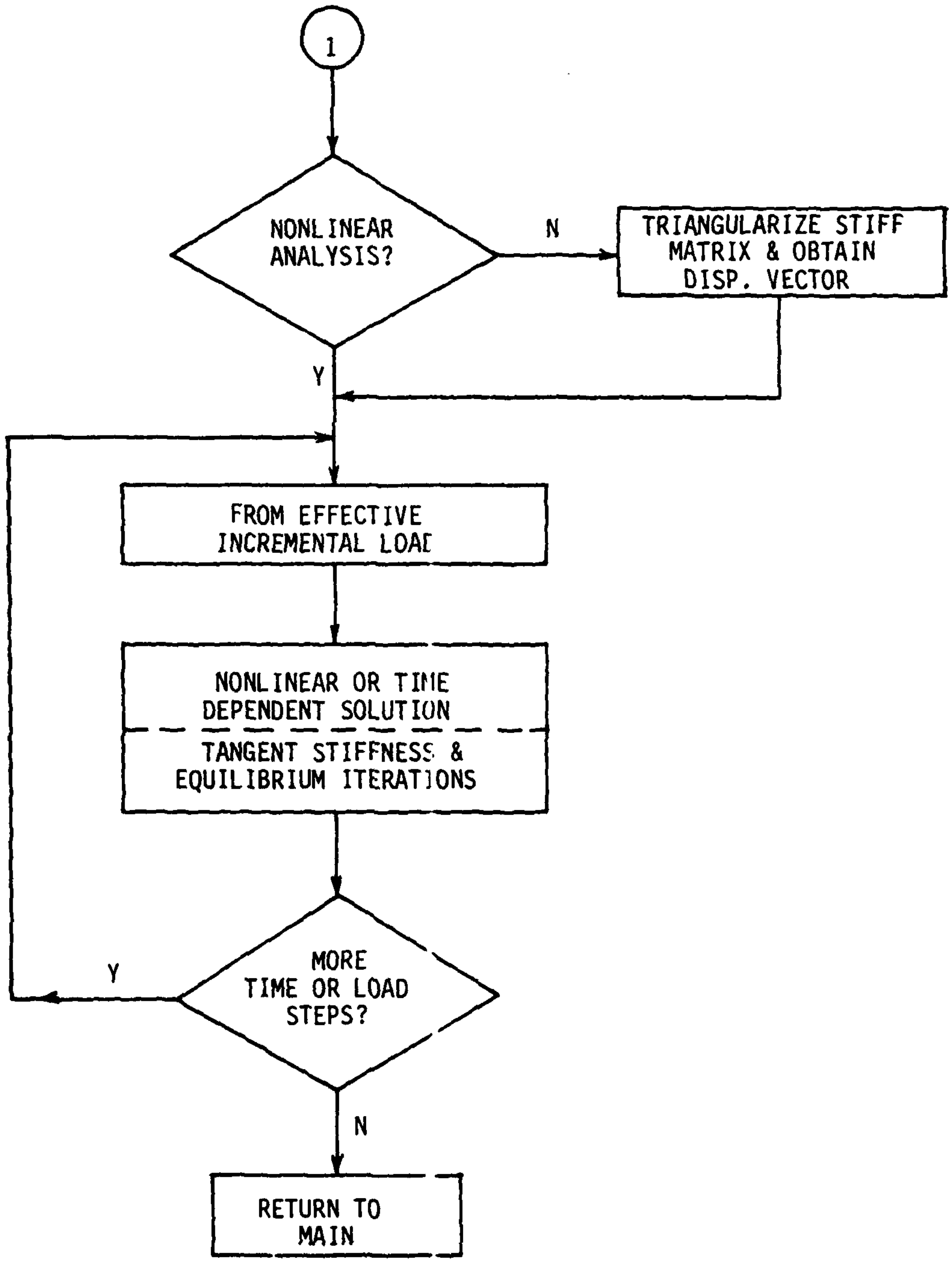


Fig. 8. (Continued)

Table 6. Macro Command for Linear Static Analysis

Module	Description
ADDRES	Determine the locations of diagonal elements of global stiffness
SBLOCK	Determine block number and size for in-core or out-of-core solution
ASSEM	Calculate element stiffness and assemble global stiffness
ADINI	Construct load vectors
WRITE	Print out initial displacements
RSTART	For a restart job, read displacement vector from tape #8 (optional)
COLSOL	Triangularize linear stiffness matrix
Loop (1) COLSOL	Obtain solution by load reduction and back substitution
NEWDAV	Update nodal displacements
WRITE	Print out nodal displacements
STRESS	Calculate element strains and stresses at Gauss points
RSTART	Write displacement vector on tape #8 for a restart job (optional)

(1) Loop for multiple load steps.

Table 7 Macro Command for Linear Dynamic Analysis and Frequency Analysis

Module	Description
ADDRES	(1)
SBLOCK	(1)
ASSEM	Calculate element stiffness and mass matrices Form effective global stiffness, mass and damping matrices.
ADINI	(1)
WRITE	Print out initial displacements, velocities and accelerations
RSTART	For a restart job, read displacements, velocities and accelerations from tape #8 (optional)
ASSEM	Calculate frequencies and mode shapes (2)
COLSOL	(1)
LOADEF	Form effective load vectors due to dynamic effects
COLSOL	(1)
NEWDAV	Update nodal displacements, velocities and accelerations
WRITE	Print out nodal displacements, velocities and accelerations
STRESS	(1)
RSTART	Write displacements, velocities and accelerations on tape #8 for a restart job (optional)

Time Loop

(1) Same as Table 7

(2) For frequency analysis option

Table 8 Macro Command for Nonlinear Static Analysis

Module	Description
ADDRES	(1)
SBLOCK	(1)
ASSEM	Perform the same calculations as in Table 7 for linear element group
ADINI	(1)
WRITE	(1)
RSTART	(1)
ASSEM	Calculate tangent element stiffness and internal nodal forces. Assemble global stiffness and form incremental load
COLSOL	Obtain incremental displacements
EQUIT	Perform equilibrium iterations
NEWDAV	(1)
WRITE	(1)
STRESS	(1)
RSTART	(1)

Load Increment Loop

(1) Same as Table 7

Table 9. Macro Command for Nonlinear Dynamic Analysis

Module	Description
ADDRES	(1)
SBLOCK	(1)
ASSEM	Form mass and damping matrices Form stiffness matrix for linear element group
ADINI	(1)
WRITE	(2)
RSTART	(2)
LOADEF	(2)
ASSEM	Calculate tangent element stiffness and internal nodal force Assemble global stiffness and incremental load
COLSOL	(3)
EQUIT	(3)
NEWDAV	(2)
WRITE	(2)
STRESS	(1)
RSTART	(2)

Time Loop

(1) Same as Table 7

(2) Same as Table 8

(3) Same as Table 9

It is noted that for the static analysis the module LOADEF is still entered and then immediately returned without performing any calculations. Another point of difference is that the dynamic analysis is repeated over a time loop as indicated in Table 7 whereas the static analysis may loop over the multiple load steps. By comparing Tables 8 and 9 with Tables 6 and 7, the major differences between the nonlinear and linear analyses (for both static and dynamic) are:

1) Formation of tangent stiffness in ASSEM
2) Performing equilibrium iterations in EQUIT

In conjunction with the aforementioned macro command, the central memory for data storage and work space is utilized by defining a master array in the blank common block. This master array is further divided sequentially into subregions for arrays which have variable lengths (also called dynamic allocations) depending on the storage requirements of the modules involved. In the main driver AAMAIN, array sizes are defined prior to the analysis modules that are called and hence the pointer for dynamic allocations vary from one stage of the analysis to the other. Because of this relationships, we may draw the following characteristic statements:

1) The procedure library is commanded in AAMAIN by use of control variables.
2) Array allocations in the blank common block vary with the algorithm procedures and the manner of partitioning the master array for various analysis steps are interrelated.

3) Coding organization of the procedure library does not conform to the modular concept. Consequently, it is rather difficult to alter this part of the coding in ADINA without upsetting the existing analysis procedures.

For completeness, the flow diagrams for the following computations are also included in the Appendix A.

A4 - Frequency Analysis

A5 - Calculation of Effective Load

A6 - Assembly of Structural Stiffness

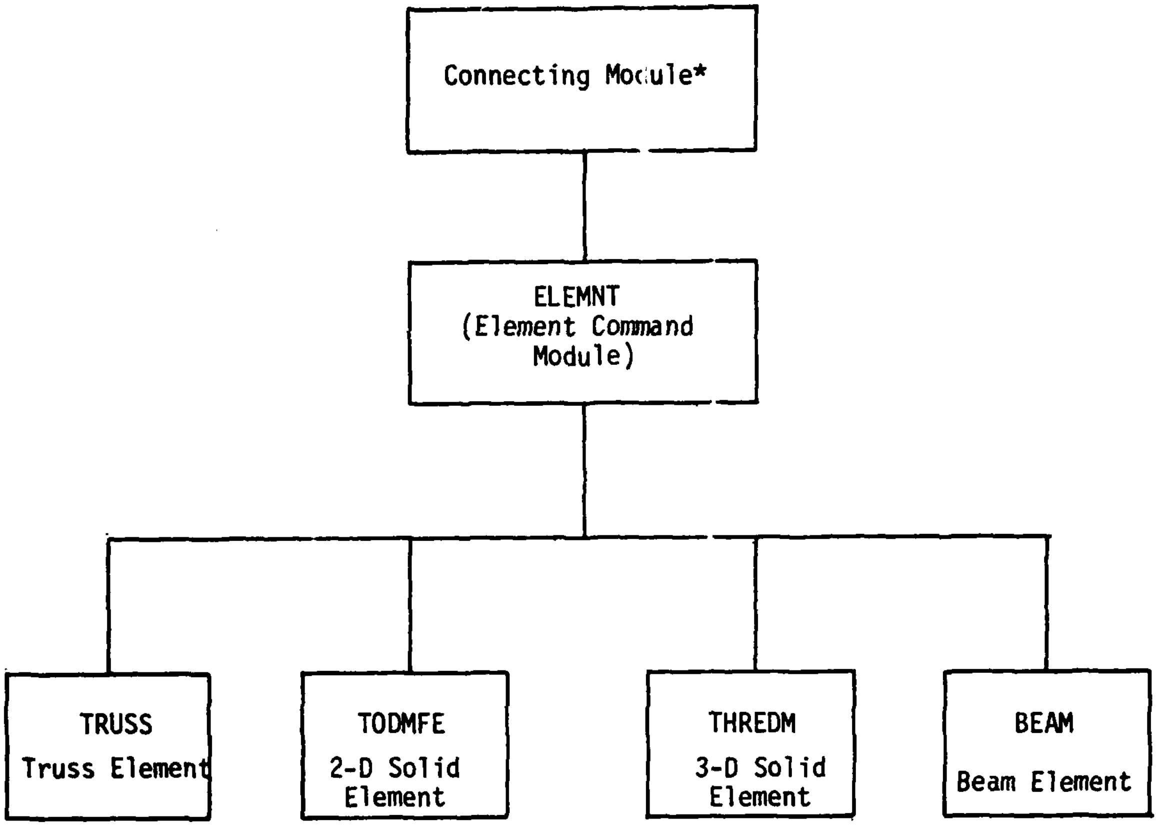
A7 - Equilibrium Iteration

A8 - Stress Calculation

2.2 Geometric Library

The Geometric library is composed of various element type which are used to model the geometric configuration and response of a structure under given loads. This library is commanded by the routine "ELEMNT" with ELCAL or ASSEM being a connecting routine between ELEMNT and the main driver AAMAIN as shown in Fig. 9. The routine ELEMNT then calls for four element control rountines, i.e.

1) TRUSS - Truss element
2) TODMFE - 2/D Continuum element
3) THREDM - 3/D Continuum element
4) BEAM - Beam element



* ELCAL - for input of element data.

ASSEM - for element stiffness or load calculation.

Fig. 9. Element Library

The flow diagrams for the above element packages are further illustrated in sections A9 - A12 of the Appendix A. Each of the above routines dynamically allocates the in-core storage space for data processing of that element. Then the element control routine calls for the lower level routines to perform the following functions:

1) Read and generate element definitions and material properties.
2) Form the linear element stiffness matrix.
3) In case of dynamic analysis, form the lumped (or consistent) mass matrix.
4) Form the nonlinear (material or/and geometric) stiffness matrix and internal nodal force vector.
5) Calculate element strains and stresses and print out these quantities.

All the above functions are coded in the sub-module, e.g. TDFE for 2/D continuum element. The calling sequence of the element module to perform the aforementioned function is shown in Table 10. Execution decision on these functions are made by checking the present values of two control variables:

IND - Flag for analysis phase

KPRI - Flag for strain and stress calculations

Among the four element types included in ADINA, the coding for the 2/D element (TDFE) and 3/D element (THDFE) was written in a parallel pattern. However, the coding for the truss and beam elements does not appear to be as well organized as the continuum elements.

Table 10. Calling Sequence of the Element Module

Analysis Stage	Calling Routines	Description
Input	ELCAL	Read element data and material properties
Formation of Element Matrices	ASSEM	Formation of element stiffness and mass matrices, and calculate internal nodal force vector.
Equilibrium Iteration	EQUIT	Calculate internal nodal forces for nonlinear elements.
Stress Calculations	STRESS	Calculate element strains and stresses

One important characteristics of the element module is that each element type is a self-contained unit which is completely independent from other elements. Therefore, modification or addition of an element to the program is a relatively simple task.

2.3 Material Library

As noted previously, the material library in ADINA was coded as a subsidiary of the element module. That is, each element type has its own list of material models. For the 2/D and 3/D continuum elements, the material models are separated into two distinct groups:

1) Linear models - controlled by STSTL (2/D element) or STST3L (3/D element).
2) Nonlinear models - controlled by STSTN (2/D element) or STST3N (3/D element).

However, for truss and beam elements, coding of the material matrix are interspaced in the element routines, which make the future program changes less apparent.

As indicated in the Appendix C of ADINA user manual [2], implementation of user-supplied nonlinear material model is possible. Although provisions have been made inside the program to accomodate new material model for 2/D and 3/D elements, such implementation is by no means obvious. The difficulty arises from the dynamic allocation of working storage that is required in the material subroutine. A clear description on the storage allocation was not given in the manual and therefore the user must figure it out from the existing material subroutines.

Since the constitutive library was organized as a lower level module of the element library, it is therefore element-dependent. The obvious advantage of such coding structure is that the logical data of the material points can be transferred directly to the element stiffness calculations. Moreover, the fast core storage for material data processing can be more efficiently utilized without resorting into I/O operations. However, this advantage is offset by the following shortcomings:

1) Codings of the same constitutive model for different element types are repetitive.
2) The availability of constitutive models in ADINA is not uniform for all element types.

2.4 Output

The output capability of ADINA is quite simplistic, it only means the printing of nodal displacements, velocities and accelerations (in case of dynamic analysis), element strains and stresses at Gauss points. No post-processing capability is available, although it is highly desirable for the large volume of data generated from a nonlinear analysis.

The output is controlled by two modules:

WRITE - Printing of nodal displacement, velocities and accelerations

STRESS - Calculating element strains and stresses at Gauss points and printing them out.

The print-out options can be either suppressed or made at designated time blocks according to the input instruction.

3. Use of Control Variables

For a given structural problem, the analysis decision, selection of geometric and material models are determined in ADINA by use of conditional statements checking the values of control variables. If one wishes to understand the macro flow of the program, it is necessary to know the precise meaning and the assigned values of these variables during different solution stages.

The control variables used in ADINA can be grouped into four categories:

1) General Type - Variables to determine solution mode, stress calculations and print-out, time history count, etc.
2) Analysis Procedures - Variables to determine static or dynamic, linear or nonlinear analysis option.
3) Dynamic Analysis Detail - Variables to determine the type of mass system and integration operator to be used.
4) Stiffness Reformation & Iterations - Variables to determine whether the stiffness reformation or equilibrium iteration is desired.

In accordance with the above categories, definitions of the control variables, subroutines in which they are referenced, are listed in Tables 11 - 14.

Table 11. Control Variables of General Type

Name	Definition	Subroutine Referenced
KPRI	Flag for stress calculation stage	AAMAIN
	= 0 Yes, calculate element stresses[1]	TDFE[2], ELPAL[3]
	= 1 No stress calculation	
KSTEP	A counter to determine the number of load (or time) step	AAMAIN, ASSEM, BLKCNT
KTR	Flag indicating calculation stage performed in equation solver "COLSOL"	AAMAIN COLSOL
	= 1 Perform matrix triangularization and vector solution[4]	
	= 2 Perform vector solution only.	
MODEX	Flag indicating solution mode	AAMAIN, ADINI, ASSEM,
	= 0 Data check only	EQUIT, LOADEF, NEWDAV,
	= 1 Normal execution	RSTART, WRITE
	= 2 A restart job	
MODEL	Flag for determining material model number	ELCAL, TODMFE[2], STSTL[3]
	= 1, 2, . . ., etc.	STSTN[3], INITWA[3], ELPAL[3]
NPAR1	Flag for selecting element type	ELCAL
	= 1 Truss element	ELEMNT
	= 2 2/D Continuum element	
	= 3 3/D Continuum element	
	= 4 Beam element	

Notes: 1. The stress calculation option is superceded by an input variable IPS. If IPS = 0, no stress calculation is desired.

2. The flag is also used in other equivalent element routines.

3. The flag is also used in other equivalent material subroutines.

4. For linear analysis, only matrix triangularization is performed.

Table 12. Control Variables for Analysis Procedures

Name	Definition	Subroutine Referenced
IEIG	Flag indicating frequency solution = 0 No frequency solution = 1 Yes	AAMAIN, ADINI
IND	Flag to control various stages of calculations = 1 Form linear stiffness matrix = 2 Form mass, damping and effective stiffness matrices = 3 Perform frequency analysis = 4 Calculate nonlinear element stiffness and effective load vector	AAMAIN, ADIDI, ASSEM, TODMFE[1], TDFE[1], QUADS[1] ELT2D3[2]
ISTAT	Flag indicating static or dynamic analysis = 0 Static analysis = 1 Dynamic analysis	AAMAIN, ADINI, ASSEM, EQUIT, INITAL, LOADEF, NEWDAV, RSTART, WRITE
KLIN	Flag indicating linear or nonlinear problem = 0 Linear Problem = 1 Nonlinear Problem	AAMAIN, ADINI, ASSEM, ELCAL, LOADEF, NEWDAV, RSTART
INDNL	Flag indicating linear or type of nonlinear analysis (an element level flag) = 0 Linear analysis = 1 Material nonlinearity with small deformation = 2 TL formulation large deformation = 3 UL formulation large deformation	ELCAL, TDFE[1] QUADS[1], STSTN[2]
IDEATH	Flag indicating element birth and death option (an element group flag) = 0 Elements are active = 1 Elements have birth option = 2 Elements have death option	ELCAL, TODMFE[1], TDFE[1]

Notes: 1. The flag is also used in other equivalent element routines.

2. The flag is also used in other material subroutines.

Table 13. Control Variables for Dynamic Analysis Detail

Name	Definition	Subroutine Referenced
IMASS	Flag indicating type of mass system = 0 No mass effect = 1 Lumped mass matrix = 2 Consistent mass matrix[1]	AAMAIN, ADINI, ASSEM, LOADEF, TDFE[1]
IDAMP	Flag indicating nodal damping effect = 0 No damping effect = 1 Yes	ADINI, ASSEM
IOPE	Flag indicating time integration operator = 0 Default set to "2" = 1 Wilson operator = 2 Newmark operator = 3 Central difference operator	AAMAIN, ADINI, ASSEM, INITAL, LOADEF, NEWDAV, RSTART.

Note: 1. Flag is also used in other equivalent element routines.

Table 14. Control Variables for Stiffness Reformation and Iterations

Name	Definition	Subroutine Referenced
IREF	Flag indicating reformation of stiffness matrix[1] = 0 Yes = 1 No	AAMAIN, ASSEM, BLKCNT, TDFE[2]
ICOUNT	Flag indicating whether calculation is during iteration stage = 2 No = 3 Yes	AAMAIN, EQUIT, TODMFE[2], TDFE[2], QUADS[2], ELPAL[3]
IEQREF	Flag indicating convergence status in iterations = 0 Yes, solution has converged = 1 No, solution does not converge.	AAMAIN, EQUIT
IEQUIT	Flag indicating whether iterations are to be performed = 0 Yes = 1 No iterations	AAMAIN, BLKCNT

Notes: 1. Stiffness reformation may be executed every time (load) step, or any specified number of time steps. The value of IREF is reassigned in subroutine BLKCNT.

2. Flag is also used in other equivalent element routines.

3. Flag is also used in other material subroutines.

Commentary:

1) The most frequently referenced control variables are "IND, ISTAT, KLIN, INDNL, KSTEP, MODEX, MODEL". These are the major parameters to determine the solution phases in the program.
2) Initialization of the control variables is partly assigned in the input module ADINI and partly scattered in the main driver AAMAIN; therefore it is difficult to trace the exact meaning of these variables. For example, the definition of the analysis control parameter "IND" is being changed intermittently from one solution stage to the other and the reassignment of its definition also depends on the status of control variables IEIG and MODEX.
3) As seen in Table 12, use of the parameters IEIG vs. ISTAT and KLIN vs. INDNL is repetitive. The four parameters can be consolidated to reduce several conditional statements in the code.

4. Data Structure

Typical to a finite element software, large quantities of data are being processed and transferred from one module to another in ADINA. This is done by use of both fast core memory and secondary (low speed) storage. The fast core is utilized by defining a dynamically allocated storage array in the blank common block as well as fixed-dimension arrays. For the secondary storage, both sequential files and random (direct) access files are used. No clear-cut data base design is made in the code, nor is there any formal data management system, which is essential for large scale modern softwares [70,71].

4.1 Fast Core Memory

Two types of in-core array allocations were made in ADINA:

1) Data arrays with fixed lengths (or dimensions)

2) A variable array in blank common block which is randomly addressable.

Only a small portion of the core memory is blocked into fixed dimension arrays. Some of these arrays are used for data transfer through labeled common blocks between lower level modules whereas other arrays are defined locally by fixed dimension statements.

Most part of the fast core is effectively utilized by defining a master array in the blank common block which has variable length. This array is then partitioned dynamically into subregions for the data processing and data storage in various modules. The master array is primarily used as the work space and therefore, it is scratchable. Data are being transferred between the core and disk or tape files independently in various modules without going through a data manager. Moreover, both the real and integer numbers are stored in a mixed pattern. The reason for using dynamic allocation of arrays is that the total length and the manner in which the blank common block is partitioned vary with:

1) The size of an analysis problem.

2) Type of analysis option requested by the user.

3) Solution stage being carried out in the program.

Therefore, it is difficult to follow the relative position and status of a data array in the blank common block unless a concise storage picture is known. For this reason, the dynamic allocation of data arrays in the blank common block are outlined in the Appendix B according to the following solution phases:

B1. Input phase - nodal data and initialization of array variables.

B2. Input phase - element data.

a. Truss element.

b. 2/D continuum element.

c. 3/D continuum element.

d. Beam element.

B3. Addresses of diagonal elements of structural stiffness matrix.

B4. Determination of blocks for structural stiffness matrix.

B5. Assembly of constant structural matrices, i.e. linear stiffness, mass and concentrated damping matrices.

B6. Construction of load vectors.

B7. Data for a restart job.

B8. Calculation of effective load vector.

B9. Assembly of tangent structural stiffness (for out-of-core solution).

B10. Linear equation solver.

B11. Equilibrium iteration

B12. Static analysis or dynamic analysis with implicit-time integration.

B13. Dynamic analysis with explicit time integration.

The block diagrams for the array allocations in terms of the position of the pointer, array names and their descriptions, subroutines referenced are given in sections B1 - B13 of Appendix B.

Commentary:

1) Fast core memory is very efficiently utilized by dynamic allocation of arrays in the blank common block.

2) No central data management is available for data transfer between the modules or between the fast core and secondary storage devises. This will make the insertion or addition of new arrays into the blank common block much more difficult without over-writing the existing records.

3) The efficiency of the fast core usage is hampered by the coding of fixed length arrays in various sub-modules of the program.

4.2 Low Speed Storage

A total of 12 low speed storage units (not including the input unit #5 and output unit #6) are used in ADINA. Two of the units (#2 and #10) are random (direct) access files and the rest of the units are sequential files. Both disk and tape storage units are utilized as indicated in Table 15. A detailed description of the data files for storage units #1 - #13 (not including #5 and #6) is given in [2]. As seen from Table 15, most of the data files are of the sequential type, which, by definition, must be processed sequentially. That is, data records must be searched and sorted before they can be processed. Since the data stored on these units are fairly straightforward, use of the sequential files appears to be quite natural for data write and retrieval. Nevertheless, two drawbacks were noticed:

Table 15. Disk/Tape Utilization

Unit	Device	Description
1-13*	Disk	Data files (see Table 16)
56, 59, 60, 63	Tape	Nodal point temperatures preprocessed input data porthole files for saving nodal/element responses.

* For a restart job, unit #8 is a tape file.

1) A large number of low speed storage units are defined and consequently the program becomes heavily I/O bound. For example, if one is to analyze a large size nonlinear dynamic problem, inordinate amounts of data transfer between memory and out-of-core files are necessary which will in turn reduce the computational efficiency of the code.
2) Since a module or sub-module may read the data from or write the data to the sequential files independently, any inappropriate changes in the data structure (dynamically allocated) would over-write the data block without noticing it.

In addition to the use of sequential files, units #2 and #10 are the random access files, in which access can be made directly to any storage location by computing its address from the pre-determined control number. Unit #2 stores the nonlinear element group data, such as element connectivity array, material constants, strain and stress components, etc. (see Section B2); certain portions of these data are frequently needed during the various stages of computations whereas other portions (e.g. element stresses and strains) have to be updated. Moreover, Unit #10 stores the blocked effective stiffness matrices from which considerable effort in data retrieval is necessary due to the effect of block coupling during equation solving. Therefore, it is also a natural choice to make both units #2 and #10 as the random access files so that a portion of the records on the file can be conveniently processed according to its logical sequence.

V. SUMMARY AND CONCLUSIONS

The finite element program ADINA was reviewed and evaluated with respect to its nonlinear analysis capability and program architecture. In this part of the report, discussion was focussed on the theoretical basis, numerical approximations adopted, and programing details from a user's point of view. It is noted that this evaluation work was made on the 1977 version of ADINA.

Based on our study, some of the unique features about ADINA can be summarized:

1) The user's manual was clearly written and easy to follow. Even for a new user, the input data can be prepared in a fairly straightforward manner according to the instructions without much confusion.

2) Theoretical manual is well documented, in which the definitions of element stiffness, mass and material matrices used in the code are available for user's who wish to understand the code. Most of its theoretical development can be found from the developer's publications referenced in [2] and [3].

3) The coding practice in ADINA corresponds largely to the theory outlined in its manual.

4) Element types in conjunction with material models were organized on a modular basis. If a user wishes to add a new element or material model, this can be done without upsetting other parts of the program. However, addition of new elements must be limited to those having 6 or less than 6 degrees of freedom at each node. Otherwise, modification of the code becomes less apparent.

5) Calculations of element stiffness matrices, especially for simple material models such as linearly elastic, isotropic materials, were coded by explicit matrix multiplications. In this way, the amount of computational effort is kept minimum.

6) The fast core memory is efficiently utilized by dynamic allocations of the blank common block, which has a variable length depending on the size of the problem and type of analysis requested by the user.

7) The program is relatively machine independent (primarily for main-frames). Double precision arithemetic option for the IBM or UNIVAC computers, and necessary coding changes for various machines are included in the program by commented statements.

8) Organization of the program and data transfer between the modules were designed primarily according to the computational flow of the analysis procedures. Therefore, ADINA is a high efficency code for executions.

9) With the out-of-core linear equation solver, ADINA can be used for the analysis of large size application problems. For this purpose, the code runs more efficiently on a virtual core environment.

10) Since ADINA is a relatively "small" general purpose finite element program (e.g. compared to NASTRAN) and many commented statements are included in the program, it is well suited as a research tool for studies in structural mechanics.

Nevertheless, ADINA has its limitations. The limitations result largely from the narrow scope for which the program was developed. A summary of these limitations are given as below:

1) The greatest drawback of the program is the lack of a comprehensive pre- and post- processing package. Especially the preprocessing capability in terms of grid generation, grid plot and thorough data check is essential for the nonlinear analysis of application problems.

2) The 1977 version of ADINA has basically four different element types: 3/D truss, 2/D isoparametric continuum, 3/D isoparametric continuum, and 3/D beam. Therefore, geometric representation of the code for complex structures is somewhat limited.

3) The manner in handling large deformation problems is not complete. In fact, the current version can only be used for large displacements but small strains.

4) The incremental solution technique adopted in ADINA is a modified Newton-Raphson method, for which the tangent modulus together with equilibrium iterations is used. This method fails to handle the situation where the load-deformation response of a structure exhibits softening behavior and then suddenly becomes stiffened.

5) When ADINA is applied to nonlinear dynamic problems, there is always a question in determining the size of the time increment for obtaining convergent and accurate solutions. For application

purpose, it is necessary to include a heuristic self-adaptive scheme so that the program can determine the convergent solution increments internally without user's guesswork.

6) Coding organization of the procedure library does not conform to the modular concept. Consequently, it is rather difficult to alter this part of the coding.

7) Data transfer between the modules or between the fast core and secondary storage units is made without a structured data management. Therefore, it is difficult to follow the status of the data files or to make any changes on these data.

Due to the aforementioned limitations, ADINA is considered primarily a research tool for nonlinear finite element structural analysis. Although additional elements, material models or other features can be added to the program, it is doubtful that the code could be expanded significantly beyond its current level.

VI REFERENCES

1. Nickell, R.E., "The Interagency Software Evaluation Group: A Critical Structural Mechanics Software Evaluation Concept," Report PT-U78-0246, Pacifica Technology, Del Mar, California, 1978.

2. Bathe, K.J., "ADINA - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis," Report 82448-1, Acoustic and Vibration Laboratory, Mechanical Engineering Department, M.I.T., September 1975, (revised May 1977).

3. Bathe, K. J., "Static and Dynamic Geometric and Material Nonlinear Analysis Using ADINA," Report 82448-2, Acoustic and Vibration Laborary, Mechanical Engineering Department, M.I.T., May 1976, (Revised May 1977).

4. Bathe, K.J., Wilson, E.L. and Iding, R.H., "NONSAP - A Structural Analysis Program for Static and Dynamic Response of Nonlinear System," SESM Report No. 74-3, Department of Civil Engineering, University of California, Berkeley, 1974.

5. The NASTRAN User's Manual (Level 17.5), NASA SP222(05), December 1978, COSMIC, University of Georgia, Athens, Georgia.

6. Logcher, R. D., et al., "ICES STRUDL - II, The Structural Design Language Engineering User's Manual," Research Report R68-91, Department of Civil Engineering, M.I.T., November 1968.

7. Bathe, K.J., Wilson, E. L., and Peterson, F.E., "SAP IV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems," Report No. EERC 73-11, College of Engineering, University of California, Berkeley, June 1973.

8. Zienkiewicz, O.C., The Finite Element Method, McGraw-Hill Book Company, New York, 1977.

9. Bathe, K.J., "ADINAT - A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis of Temperatures," Report 82448-5, Acoustic and Vibration Laboratory, Mechanical Engineering Department, M.I.T., May 1977.

10. Padovan, J. and Chang, T.Y., "Evaluation of ADINA: Part I, Operating Characteristics," Report No. AUE-802, College of Engineering, University of Akron, June 1980.

11. Martin, H.C., "Finite Elements and the Analysis of Geometrically Nonlinear Problems," Recent Advances in Mat. Meth. Str. Anal. and Design (eds., R. H. Gallagher et al.), Univ. of Alabama Press, 1971, pp. 343-381.

12. Oden, J.T., Finite Elements of Nonlinear Continua, McGraw-Hill Book Company, New York, 1972.

13. Hibbitt, H.D., Marcal, P.V., and Rice, J.R., "Finite Element Formulation for Problems of Large Strain and Large Displacements," Int. J. Solids Struct., Vol. 6, 1970, pp. 1069-1086.

14. McNamara, J.F. and Marcal, P.V., "Incremental Stiffness Method for Finite Element Analysis of the Nonlinear Dynamic Problem," Numerical and Computer Methods in Structural Mechanics, S. J. Feaves, N. Perrone, J. Robinson and W.C. Schnobrich, eds., Academic Press, New York, 1973.

15. Sharifi,P. and Yates,D.N.,"Nonlinear Thermo-Elastic-Plastic and Creep Analysis by the Finite Element Method," AIAA Journal, Vol. 12, No. 9, September 1974, pp. 1210-1215.

16. Needleman, A., "A Numerical Study of Necking in Circular Cylindrical Bars," J. Mech. Phys. Solids, Vol. 20, 1972, pp. 111-127.

17. Hutchinson, J.W., "Finite Strain Analysis of Elastic-Plastic Solids and Structures," Numer. Solution of Nonlinear Struc. Problems, (ed. R.F. Hartung), ASME, 1973, pp. 17-29.

18. Bathe, K.J., Ramm, E., and Wilson, E.L., "Finite Element Formulations for Large Deformation Dynamic Analysis," Int. J. Numerical Methods in Engng., Vol. 9, 1975, pp. 353-386.

19. Argyris, J.H. and Kleiber, M.,"Incremental Formulation in Nonlinear Mechanics and Large Strain Elasto-Plasticity - A Natural Approach. Part I," Computer Methods in Applied Mechanics and Engineering, Vol. 11, 1977, pp. 215-247.

20. Yamada, H., Jirakawa, T. and Wifi, A.S., "Analysis of Large Deformation and Bifurcation in Plasticity Problems by the Finite Element Method," Proceedings on Finite Elements in Nonlinear Mechanics, Tapir Press, Norwegian Institute of Technology, Trondheim, Norway, 1977.

21. Yaghmai, S., and Popov, E.P., "Incremental Analysis of Large Deflections of Shells of Revolution," Int. J. Solids Struct., 7, 1971, pp. 1375-1393.

22. Sharifi, P., and Popov, E.P., "Nonlinear Finite Element Analysis of Sandwich Shells of Revolution," AIAA Journal, 11, 1973, pp. 715-722.

23. Hofmeister, L.D., Greenbaum, G.A. and Evenson, D.A., "Large Strain, Elastic-Plastic Finite Element Analysis," AIAA Journal, Vol. 9, No. 7, 1971, pp. 1248-1254.

24. Osias, J.R., and Swedlow, J.L., "Finite Deformation of Elasto-Plastic Deformation -I: Theory and Numerical Examples," Int. J. Solids and Struct. Vol. 10, 1974, pp. 321-339.

25. Hill, R., "Some Basic Principles in the Mechanics of Solids without a Natural Time," J. Mech. Phys. Solids, 7, 1959, pp.209-225.

26. McMeeking, R.M. and Rice, J.R., "Finite-Element Formulations for Problems of Large Elastic-Plastic Deformation," Int. J. Solids Structures, Vol. 11, 1975, pp. 601-616.

27. Yamada, Y., "Time Dependent Materials," in Computer Programs in Shock and Vibration, The Shock and Vibration Information Center, Naval Research Laboratory, Washington D.C., 1975, pp. 173-188.

28. Nemat-Nessar, S and Taya, M., "Model Studies of Ductile Fracture-I. Formulation," Franklin Institute Journal, Vol. 302, 1976, pp. 463-472.

29. Atluri, S.N., "On Some New General and Complementary Energy Theorems for the Rate Problems in Finite Strain, Classical Elasto-Plasticity," Report No. GIT-ESM-SNA-10, Georgia Institute of Technology, Atlanta, Georgia, December 1978.

30. Murakawa, H. and Atluri, S.N., "Finite Element Solutions of Finite Strain Elastic - Plastic Problems, Based on a Complementary Rate Principle," Technical Report No. 1 GIT-CE-SNA-1, Georgia Institute of Technology, Atlanta, Georgia, June 1979.

31. Nagtegaal, J.C., Parks, D.M. and Rice, J.R.,"On Numerical Accurate Finite Element Solutions in the Fully Plastic Range," Comp. Meth. Appl. Mech. Eng., Vol. 4, 1974, pp. 153-177.

32. Bleich, H.H., "On the Use of a Special Nonassociated Flow Rule for Problems on Elasto-Plastic Wave Propagation," Report No. DASA 2635, March 1971.

33. Bleich, H.H., "On Uniqueness in Ideally Elastoplastic Problems in Case of Nonassociated Flow Rules," J. Appl. Mech. December, 1972, pp. 983-987.

34. Bathe, K.J., and Ramaswamy, S., "On Three-Dimensional Nonlinear Analysis of Concrete Structures," Nuclear Engineering and Design, Vol. 52, No. 3, 1979, pp. 385-409.

35. Connor, J.J., and Sarne, Y., "Nonlinear Analysis of Prestressed Concrete Reactor Pressure Vessels," Paper presented at 3rd International Conference on Structural Mechanics in Reactor Technology, Berlin, Germany, September 1975.

36. Chen, A.C.T. and Chen, W.F., "Constitutive Relations for Concrete," J. Eng. Mech.Div., ASCE, Vol. 101, No. EM4, August 1975, pp. 465-481.

37. Bazant, Z.P., "Endochronic Theory of Inelasticity and Failure of Concrete," J. Eng. Mech. Div., ASCE, Vol. 102, No. EM4, August 1976, pp. 701-722.

38. Elwi, A.A. and Murray, D.W., "A 3D Hypoelastic Concrete Constitutive Relationship," J. Eng. Mech. Division, ASCE, Vol. 105, No. EM4, August 1979, pp. 623-641.

39. Chen, W.F. and Ting, C.E., "Constitutive Models For Concrete Structures," J. Eng. Mech. Div., ASCE, Vol. 106, No. EM1, Feb. 1980, pp. 1-19.

40. Cook, R.D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, New York, 1974.

41. Bathe, K.J. and Wilson, E.L., Numerical Methods in Finite Element Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

42. Przemieniecki, J.S., Theory of Matrix Structural Analysis, McGraw-Hill Book Company, New York, 1968.

43. Chang, T.Y., Prachuktam, S. and Reich, M., "Assessment of a Nonlinear Structural Analysis Finite Element Program (NONSAP) for Inelastic Analysis," *Paper presented at the Energy Technology Conference*, Houston, Texas, September 18-22, 1977, an ASME Paper 77-PVP-10.

44. Barsoum, R.S., "Application of Triangular Quarter-Point Elements as Crack Tip Elements of Power Law Hardening Material," *Int. J. Fract.*, 1966, pp. 463-466.

45. Bathe, K.J., and Bolourchi, S., "A Large Deisplacement Analysis of Three-Dimensional Beam Structures," Int. J. Num. Meth. Eng. Vol. 14, No. 7, 1979, pp. 961-986.

46. Gere, J.M. and Weaver, W., Jr., Analysis of Frame Structures, D. Van Nostrand Company, New York, 1965.

47. Stricklin, J.A., and Haisler, W.E., "Formulations and Solution Procedures for Nonlinear Structural Analysis," J. Computers and Structures, Vol. 7, 1977, pp. 125-136.

48. Bathe, K.J. and Ozdemir, H., "Elastic-Plastic Large Deformation *Static and Dynamic* Analysis," J. Computers and Structures, Vol. 6, No. 2, 1976, pp. 81-92.

49. Mondkar, D.P. and Powell, G.H., "Evaluation of Solution Schemes for Nonlinear Structures," J. Computers & Structures, Vol. 9, 1978, pp.223-236.

50. Newmark, N.M., "A Method of Computation for Structural Dynamics," J. Eng. Mech. Div., ASCE, Vol. 85, EM3, 1959, pp. 67-94.

51. Wilson, E.L., "A Computer Program for the Dynamic Stress Analysis of Underground Structures," Report No. SESM 68-1, Department of Civil Engineering, University of California, Berkeley, 1968.

52. Collatz, L., The Numerical Treatment of Differential Equations, Springer-Verlag, New York, 1966.

53. Leech, J.W., Hsu, P.T. and Mack, E.W., "Stability of a Finite Difference Method for Solving Matrix Equations," AIAA Journal, Vol. 3, 1965, pp. 2172-2173.

53. Krieg, R.D and Key, S.W., "Transient-Shell Response by Numerical Time Integration," Advances in Computational Methods in Structural Mechanics and Design, J.D. Oden, R. W. Clough and Yamamoto, Y., eds., University of Alabama Press, Huntsville, Ala., 1972.

55. Nickell, R.E., "On the Stability of Approximation Operators in Problems of Structural Dynamics," Int. J. Solids Structures, 7, 1971, pp. 301-319.

56. Nickell, R.E., "Direct Integration Methods in Structural Dynamics," J. Eng. Mech. Div , ASCE, Vol. 99, EM2., April 1973, pp. 303-317.

57. Goudreau, G.L. and Taylor, R.L., "Evaluation of Numerical Integration Methods in Elastodynamics," Comp. Meths. Appl. Mech. Eng., Vol. 2, 1972, pp. 69-67.

58. Bathe, D. J. and Wilson, E.L., "Stability and Accuracy Analaysis of Direct Integration Methods," Int. J. Earthquake Eng. Struct. Dyn., Vol. 1, 1973, pp. 283-291.

59. Krieg, R.D., "Unconditional Stability in Numerical Time Integration Methods," J. Appl. Mech., June 1973, pp. 417-421.

60. Park, K.C., "Practical Aspects of Numerical Time Integration," J. Computers and Structures, Vol. 7, 1977, pp. 343-353.

61. Weeks, G., "Temporal Operators for Nonlinear Structural Dynamic Problems," J. Eng. Mech. Div., ASCE, Vol. 98, No. EM5, October 1972, pp. 1086-1104.

62. McNamara, J.F., "Solution Schemes for Problems of Nonlinear Structural Dynamics," J. Pressure Vessel Tech. May 1974, pp. 96-102.

63. Belytschko, T. and Schoeberle, D.F., "On the Unconditional Stability of an Implicit Algorithm for Nonlinear Structural Dynamics," J. Appl. Mech., Dec. 1975, pp. 865-869.

64. Hughes, T.J.R., "A Note on the Stability of Newmark's Algorithm in Nonlinear Structural Dynamics," Int. J. Num. Meth. Eng., Vol. 11, No. 2, 1977, pp. 383-386.

65. Hughes, T.J.R., "Stability, Convergence and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics," J.Computers & Structures, Vol. 6, 1976, pp. 313-324.

66. Felippa, C.A. and Park, K.C., "Direct Time Integration Methods in Nonlinear Structural Dynamics," Computer Meth. Appl. Mech. Eng., 1979, pp. 277-313.

67. Argyris, J.H., Doltsinis, J. St., Knudson, W.C., Vaz, L.E. and Willam, K.J., "Numerical Solution of Transient Nonlinear Problems," Computer Meth. Appl. Mech.Eng., 1979, pp. 341-409.

68. Belytschko, T., "A Survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis," Nucl. Eng. and Design., Vol. 37, 1976, pp. 23-34.

69. Sander, G., Geradin, M., Nyssen, C. and Hodge, M., "Accuracy versus Computational Efficiency in Nonlinear Dynamics," Computer Meth. Appl. Mech. Eng., 1979, pp. 315-340.

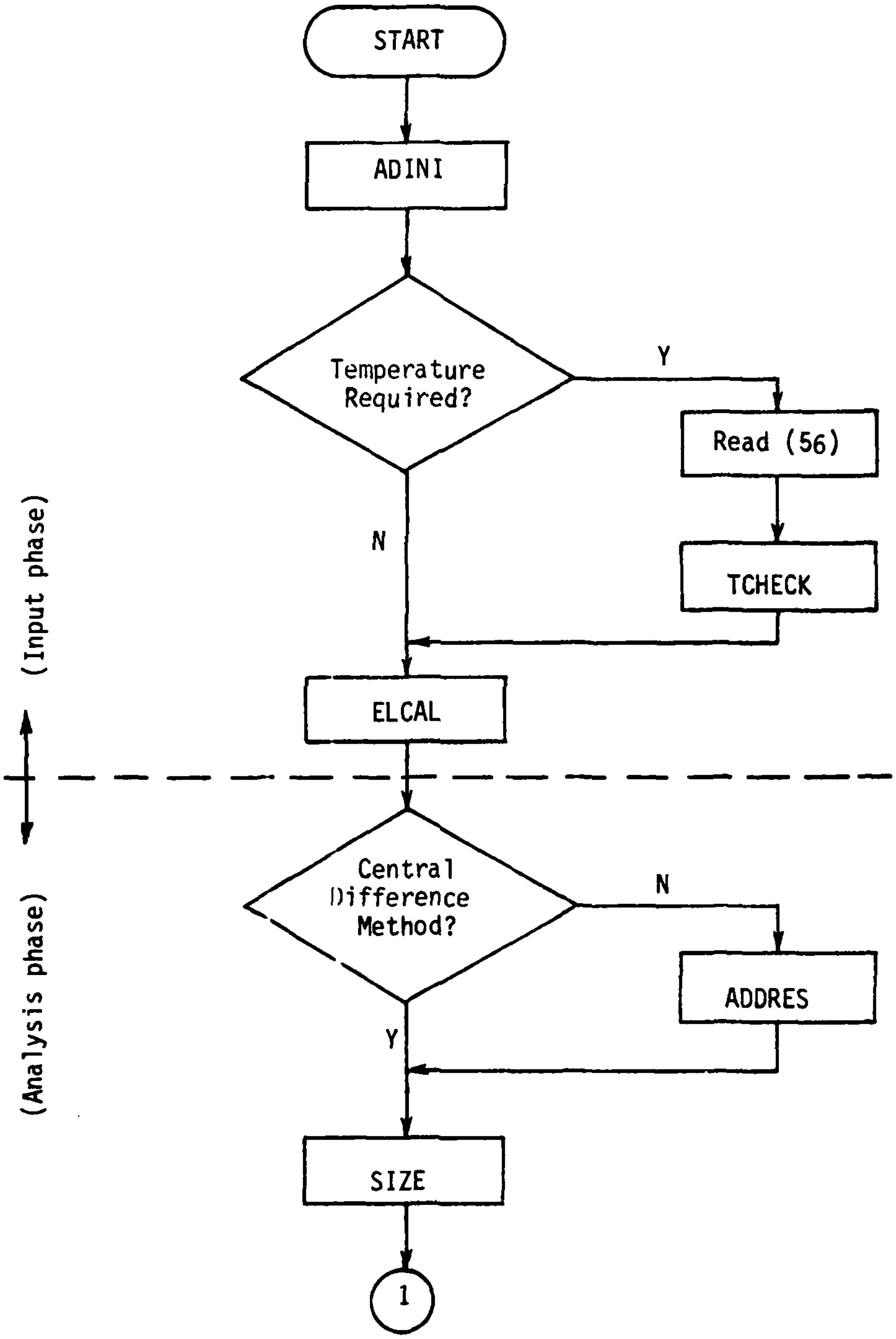
70. Felippa, C.A., "Data Base Management in Scientific Computing - I. General Description,"J.Computers & Structures, Vol. 10, 1979, pp. 53-61.

71. Dodds, R.H., Jr., Lopez, L.A. and Pecknold, D.A., "Numerical and Software Requirements for General Nonlinear Finite Element Analysis," Technical Report UILU-ENG-78-2020, University of Illinois at Urbana-Champaign, Illinois, September 1978.

Appendix A

FLOW DIAGRAMS FOR VARIOUS SOLUTION PHASES

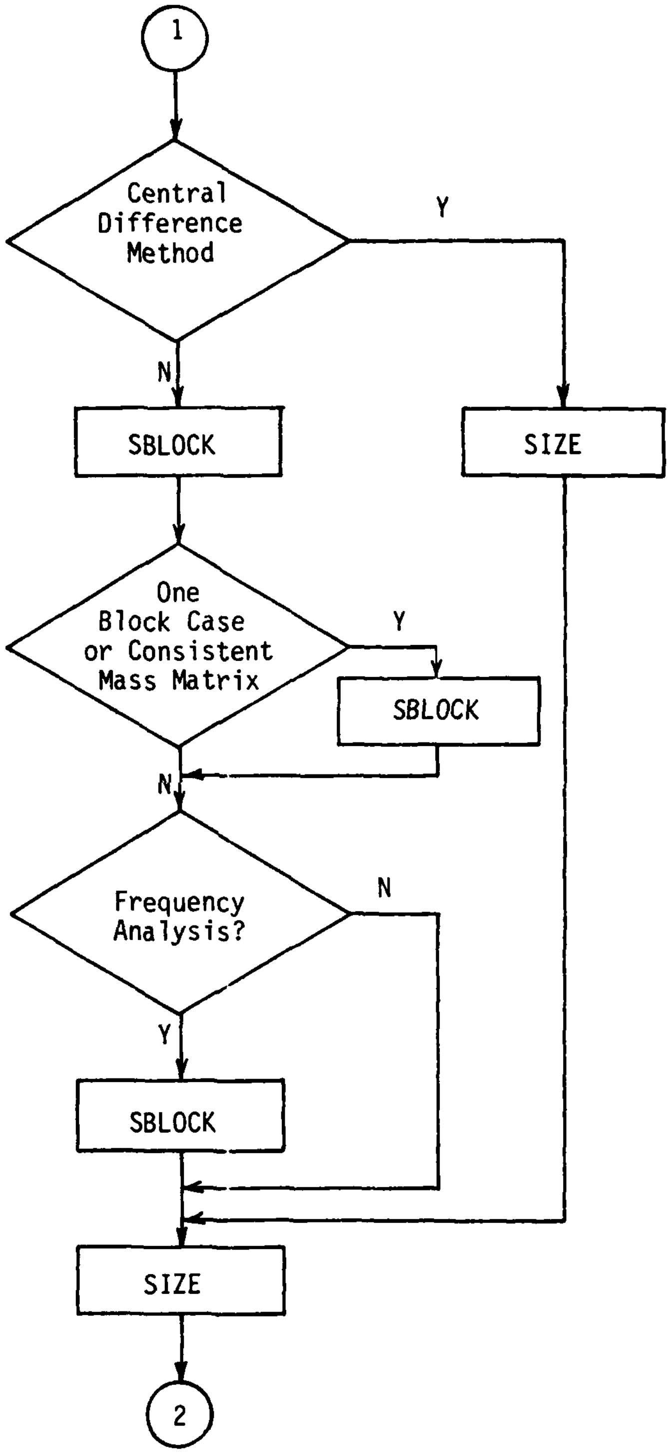
A1 Flow Diagram of the Main Program "AAMAIN"

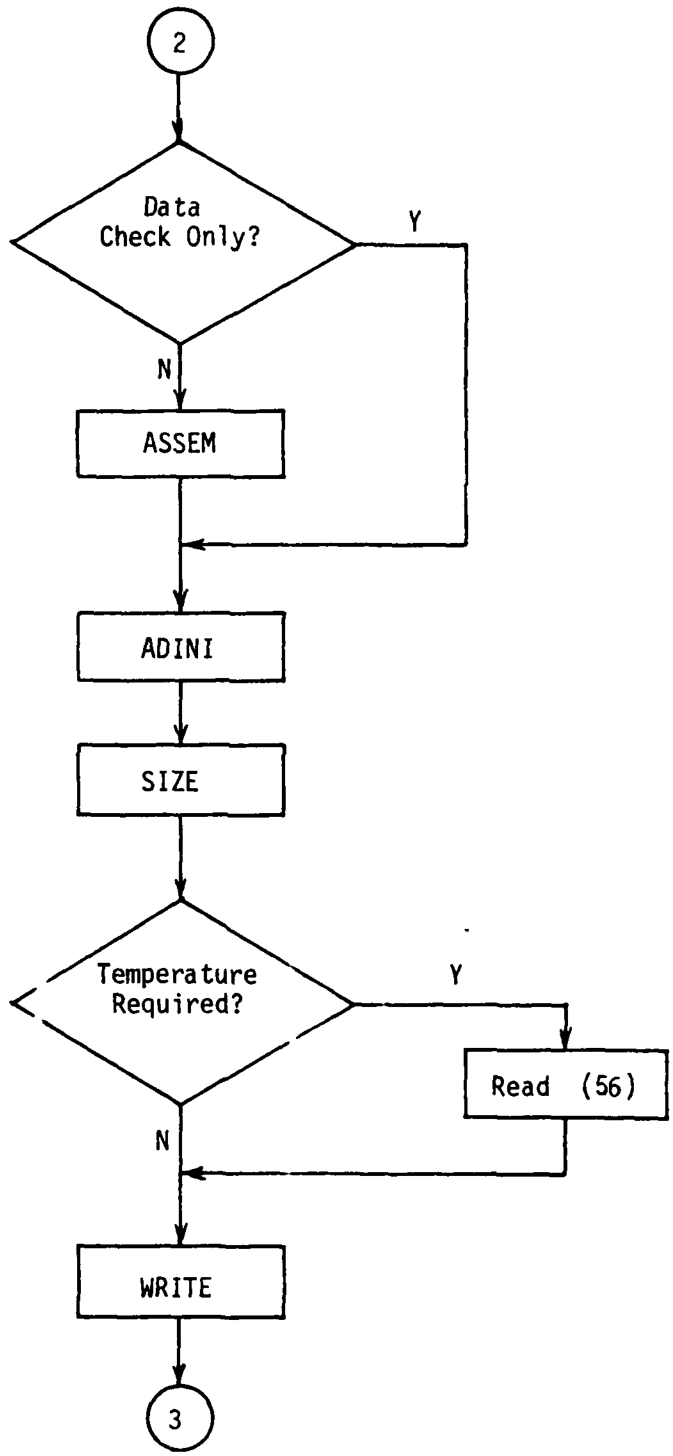


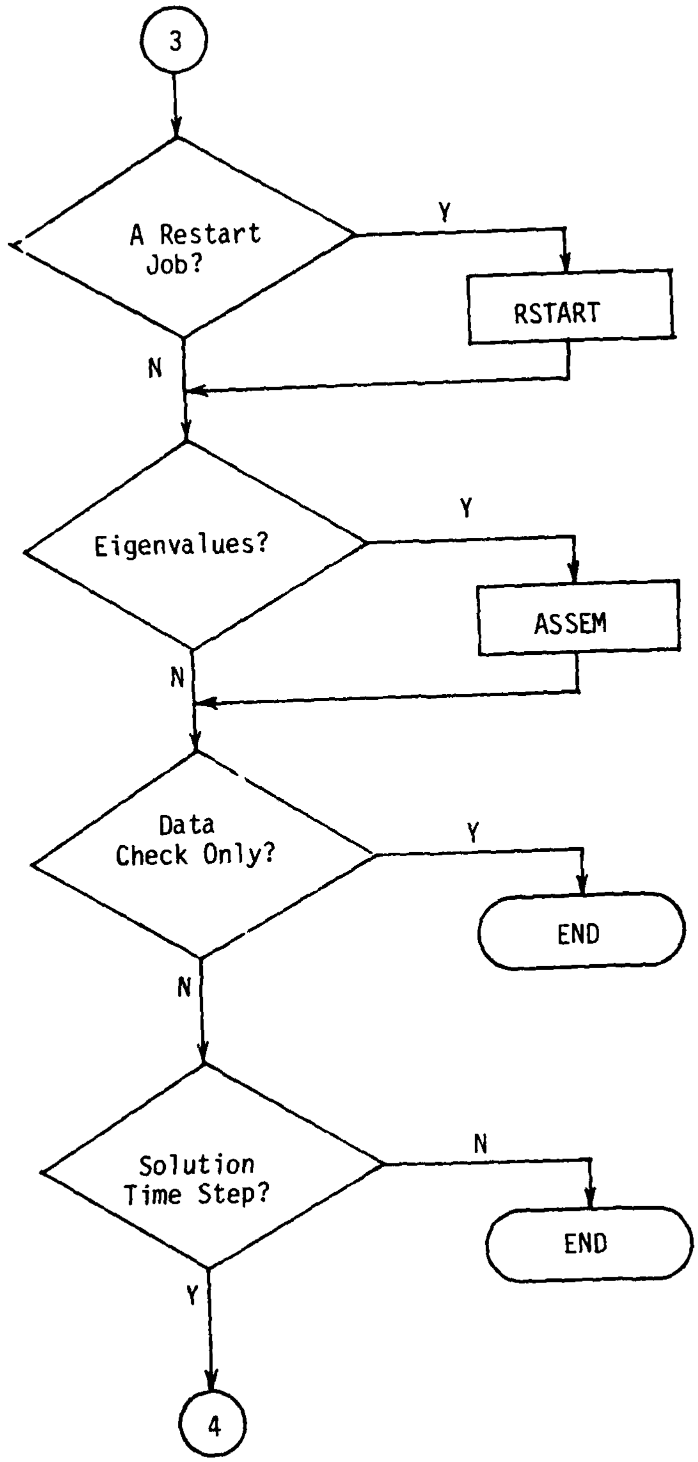
AD-A096 678 AKRON UNIV OH COLL OF ENGINEERING F/G 20/11
EVALUATION OF ADINA. PART I. THEORY AND PROGRAMING DESCRIPTIONS--ETC(U)
JUN 80 T Y CHANG, J PADOVAN N00014-78-C-0691
UNCLASSIFIED AUE-801 NL

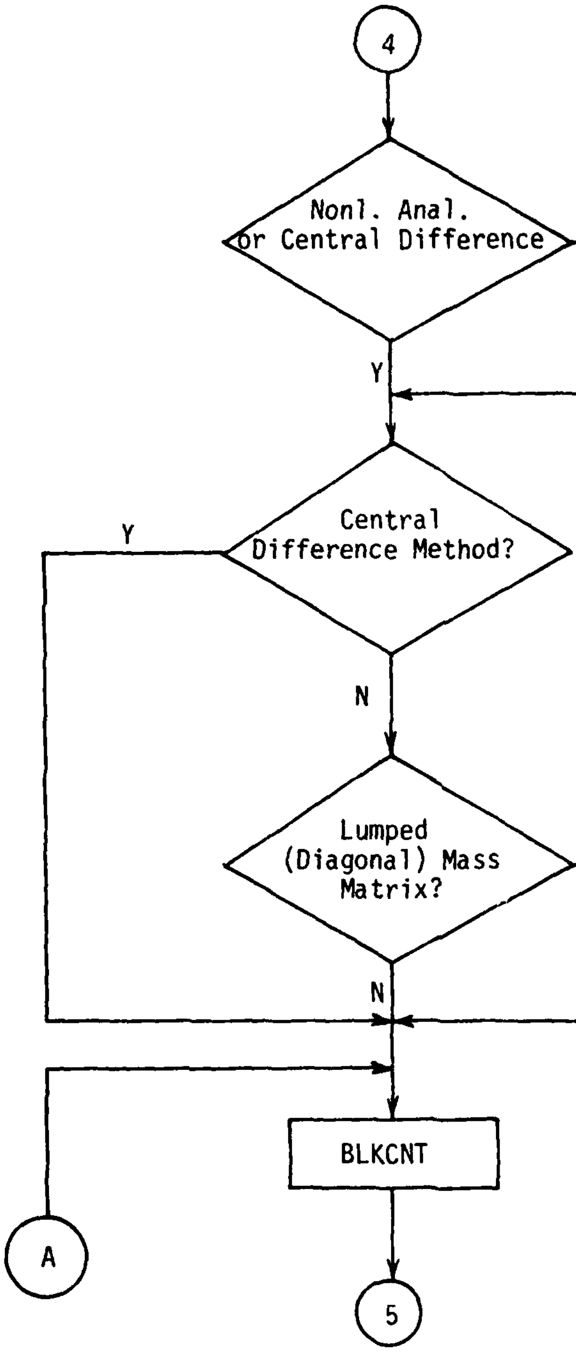
AD A096678

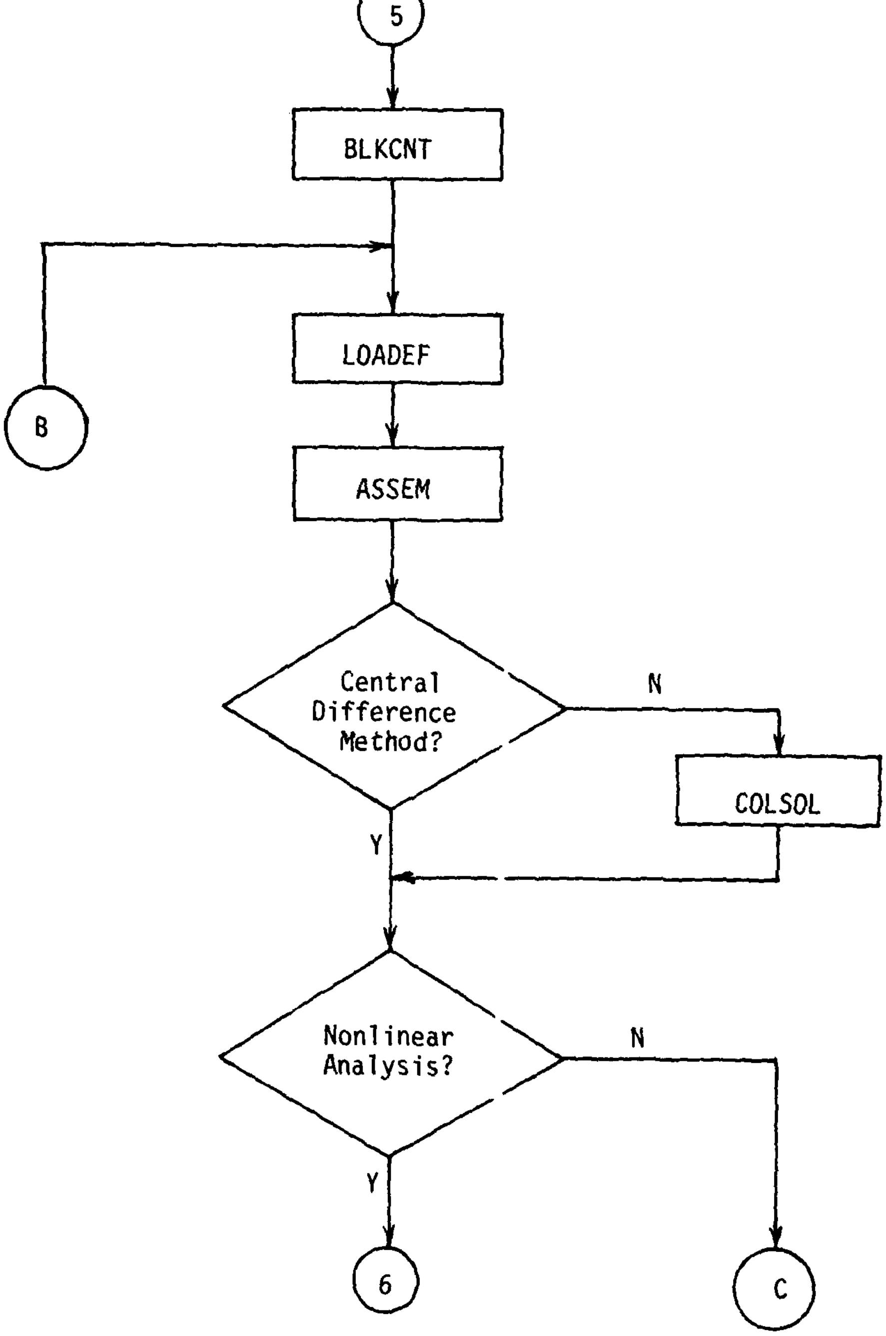
END
DATE FILMED
4-81
DTIC


1
Central Difference Method
Y
N
SBLOCK
SIZE
One Block Case or Consistent Mass Matrix
Y
SBLOCK
N
Frequency Analysis?
N
Y
SBLOCK
SIZE
2

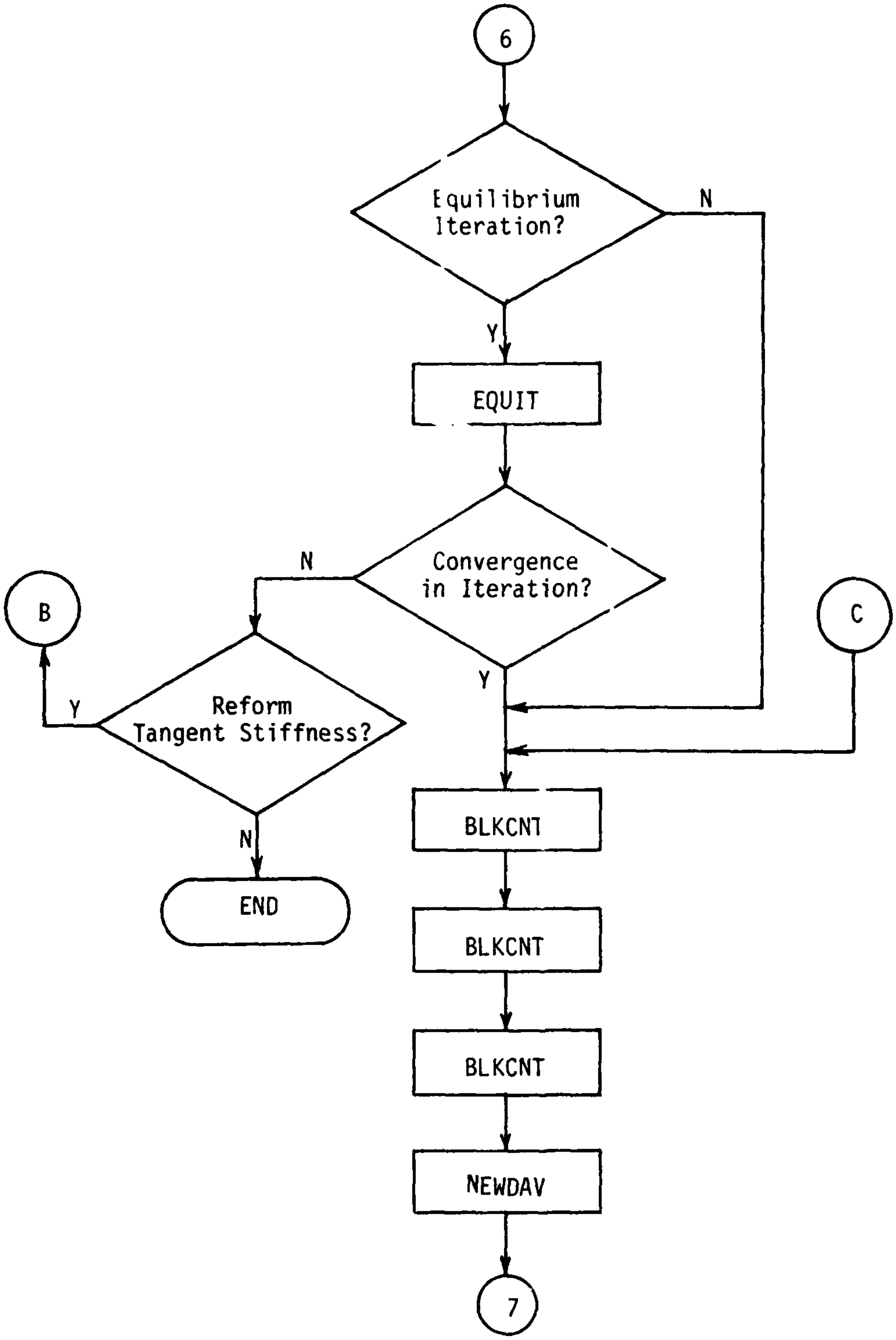

2
Data
Check Only?
Y
N
ASSEM
ADINI
SIZE
Temperature
Required?
Y
Read (56)
N
WRITE
3


3
A Restart Job?
Y
RSTART
N
Eigenvalues?
Y
ASSEM
N
Data Check Only?
Y
END
N
Solution Time Step?
N
END
Y
4

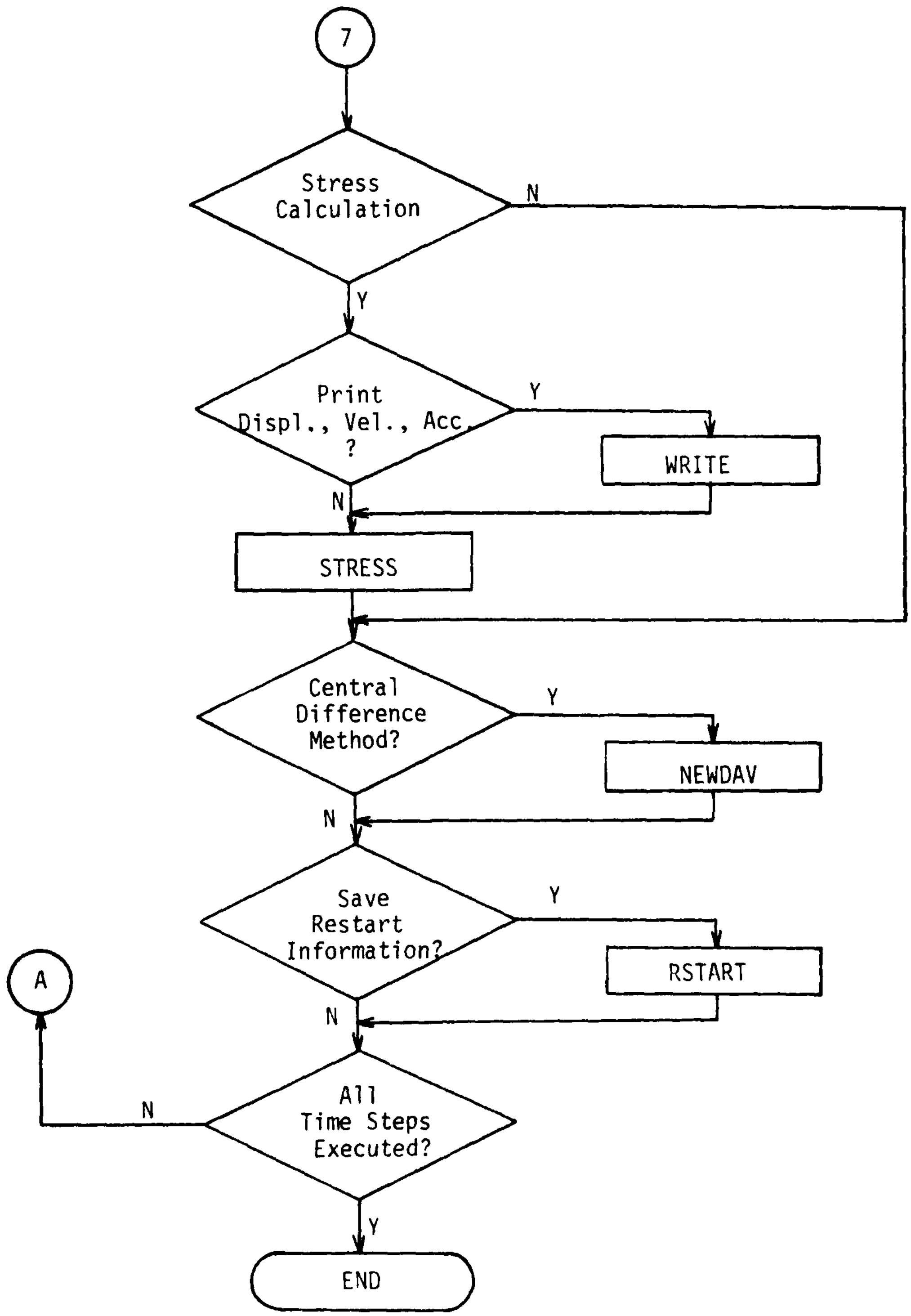

4
Nonl. Anal.
or Central Difference
N
COLSOL
Y
Central
Difference Method?
Y
N
Lumped
(Diagonal) Mass
Matrix?
Y
Read (11)
N
BLKCNT
A
5



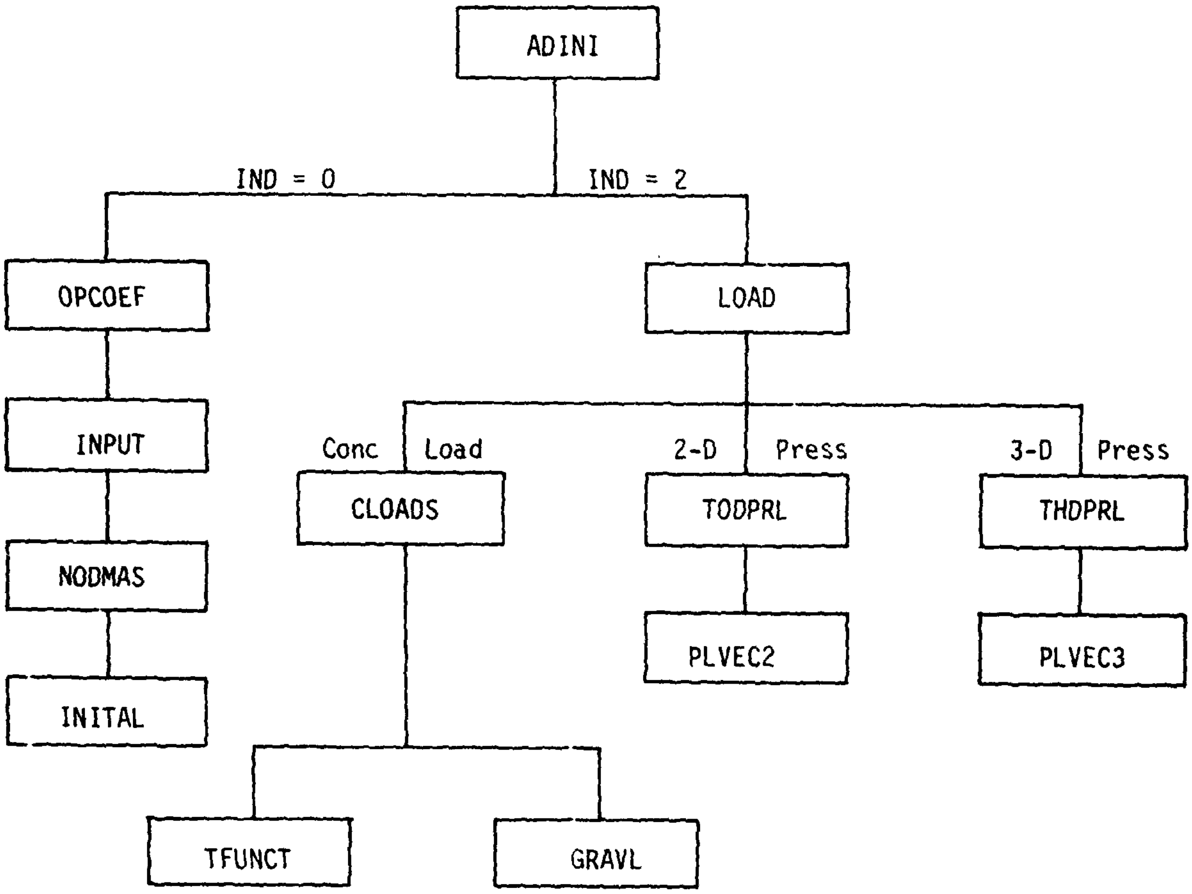
5
BLKCNT
LOADEF
B
ASSEM
Central Difference Method?
N
COLSOL
Y
Nonlinear Analysis?
N
Y
6
C



6
Equilibrium Iteration?
N
Y
EQUIT
Convergence in Iteration?
N
Y
B
C
Reform Tangent Stiffness?
Y
N
END
BLKCNT
BLKCNT
BLKCNT
NEWDAV
7


7
Stress Calculation
N
Y
Print Displ., Vel., Acc. ?
Y
WRITE
N
STRESS
Central Difference Method?
Y
NEWDAV
N
Save Restart Information?
Y
RSTART
N
A
N
All Time Steps Executed?
Y
END

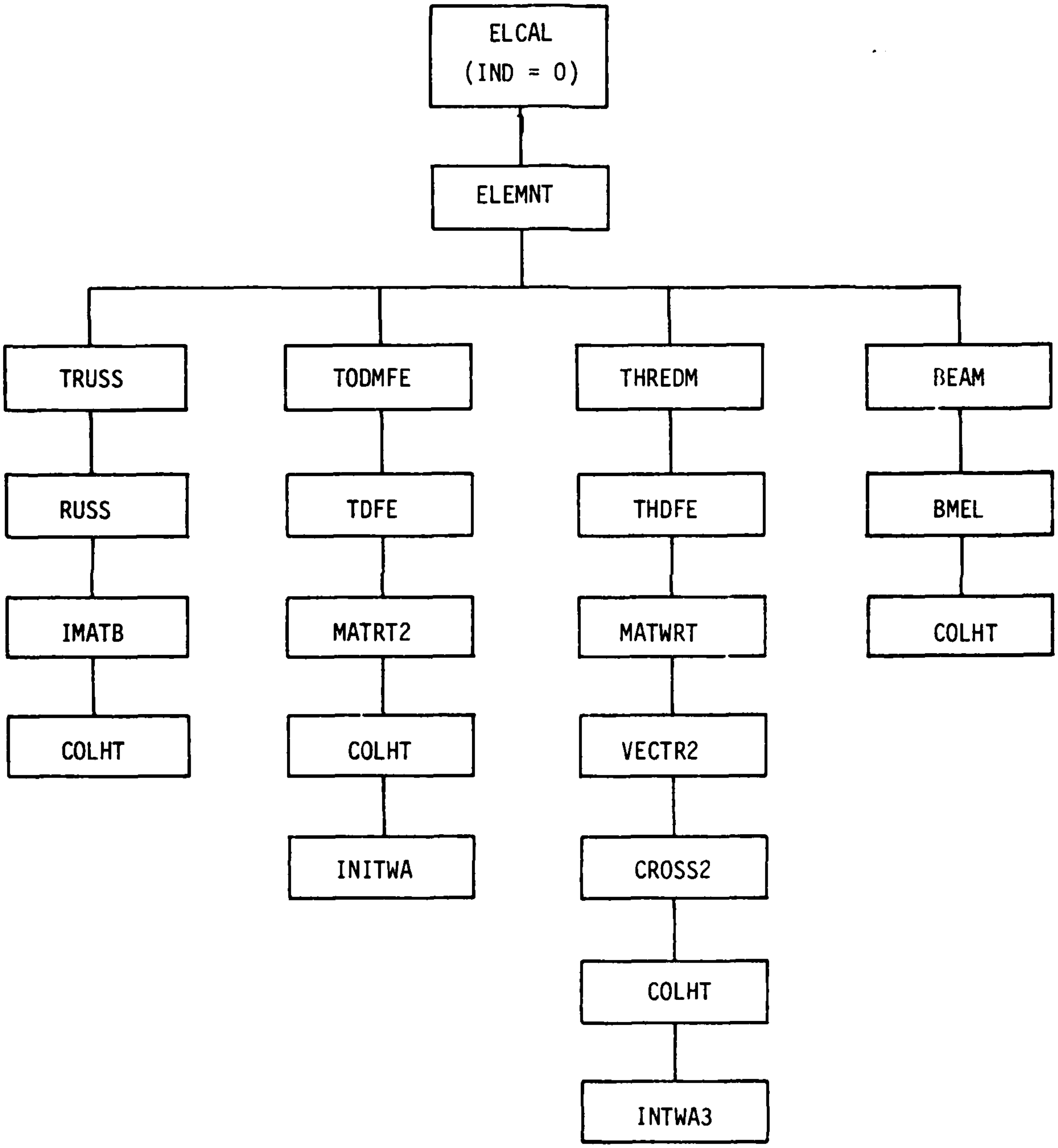
A2. Flow Diagram for Reading Input Data



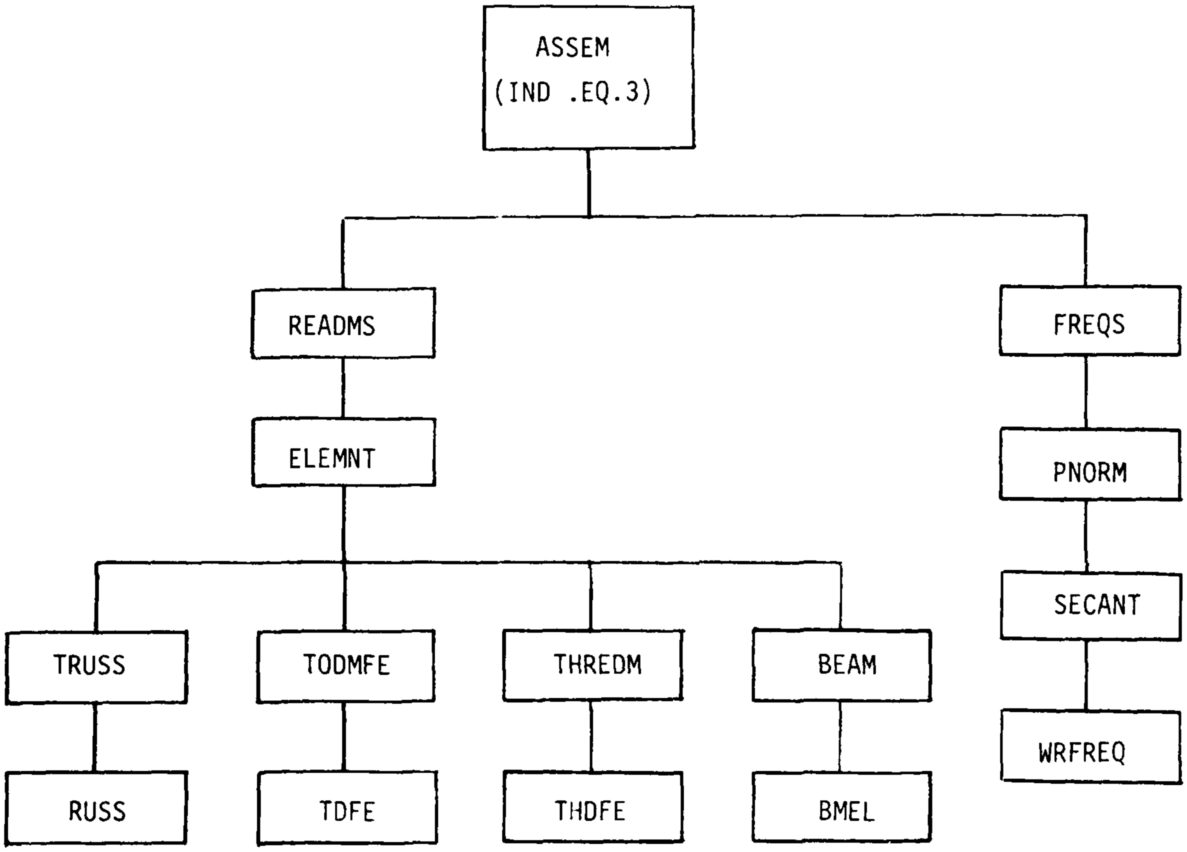
IND = 0, Input data is processed.

IND = 2, Load data is processed.

A3. Processing of Element Data

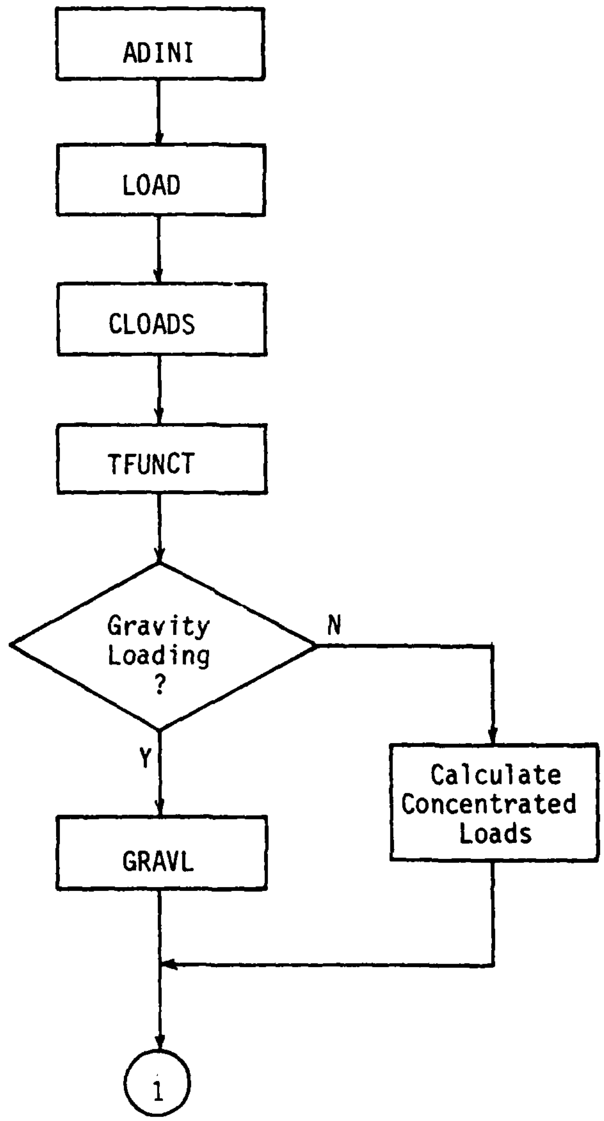


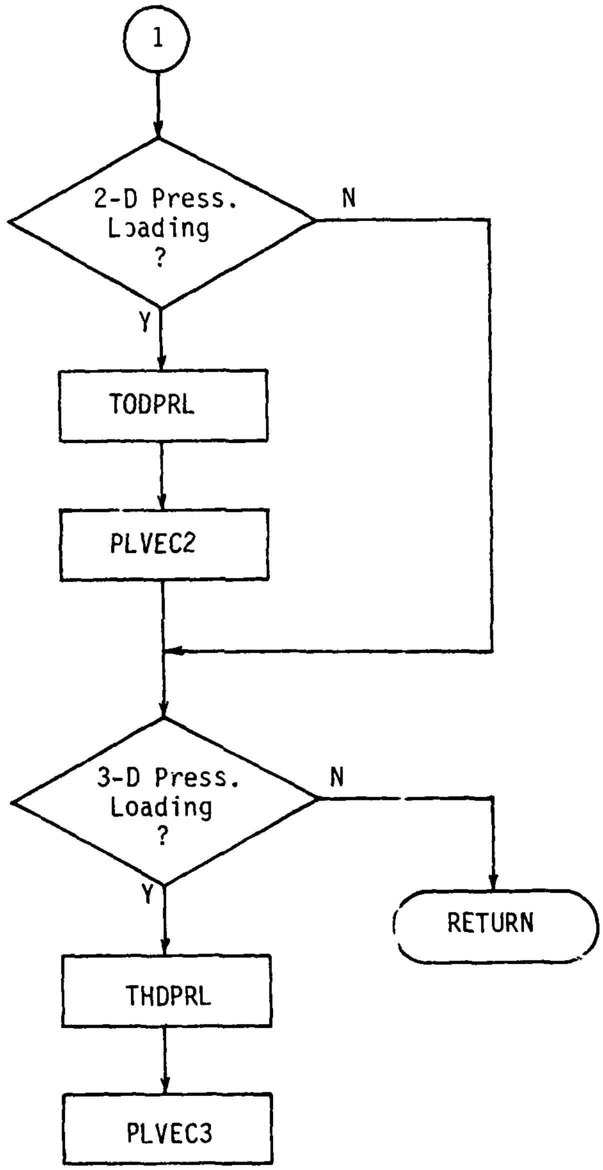
A4. Frequency Analysis



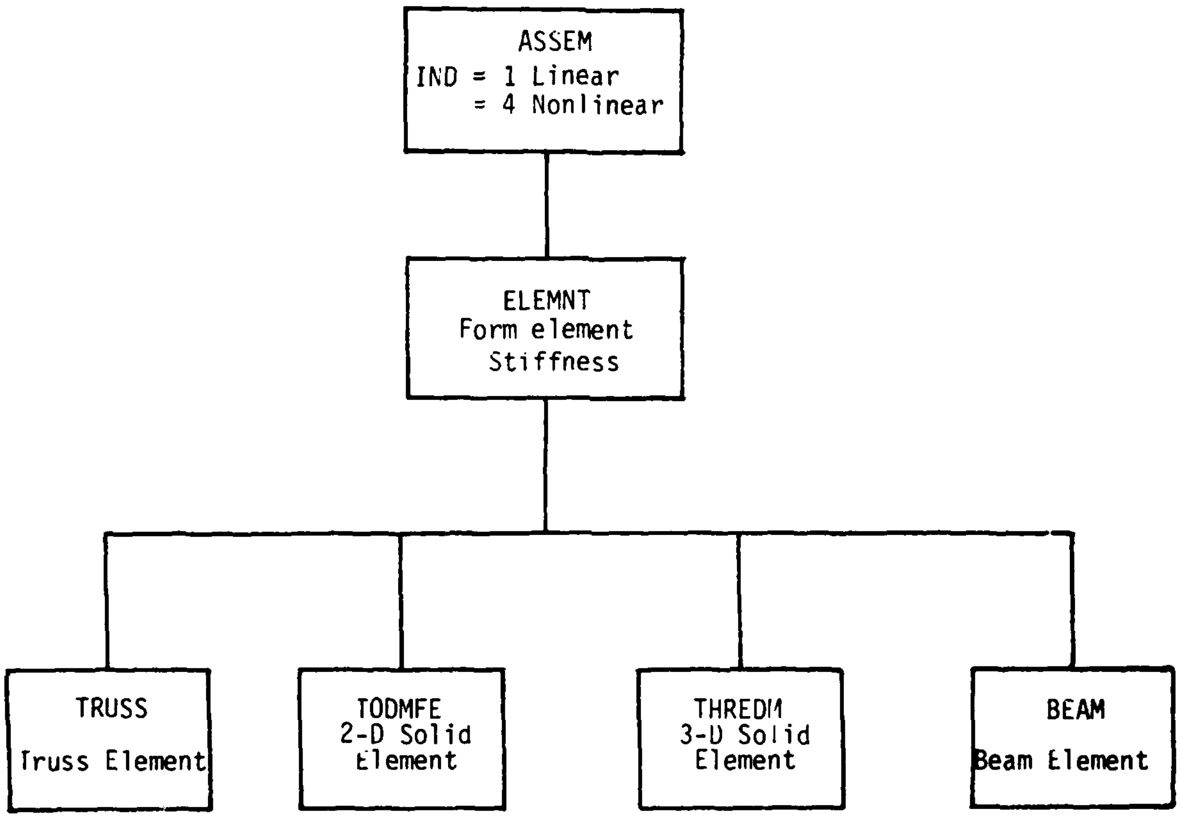
Note: When IEIG EQ.1, Frequency Analysis is requested.

A5 - <u>Calculation of Effective Load</u>

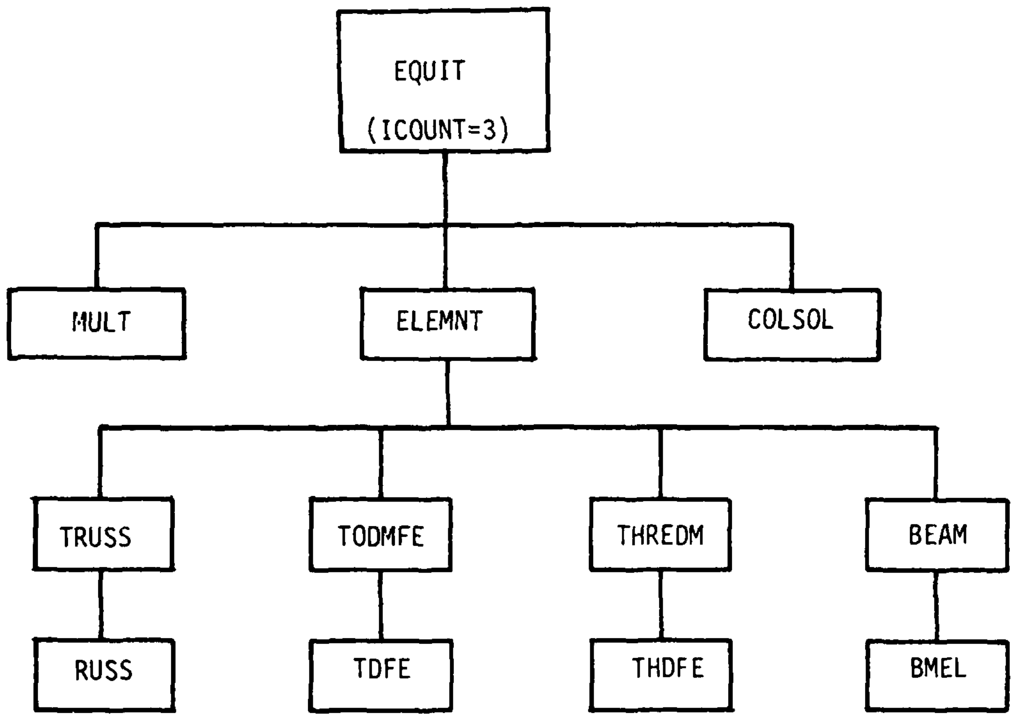



1
2-D Press.
Loading
?
N
Y
TODPRL
PLVEC2
3-D Press.
Loading
?
N
Y
RETURN
THDPRL
PLVEC3

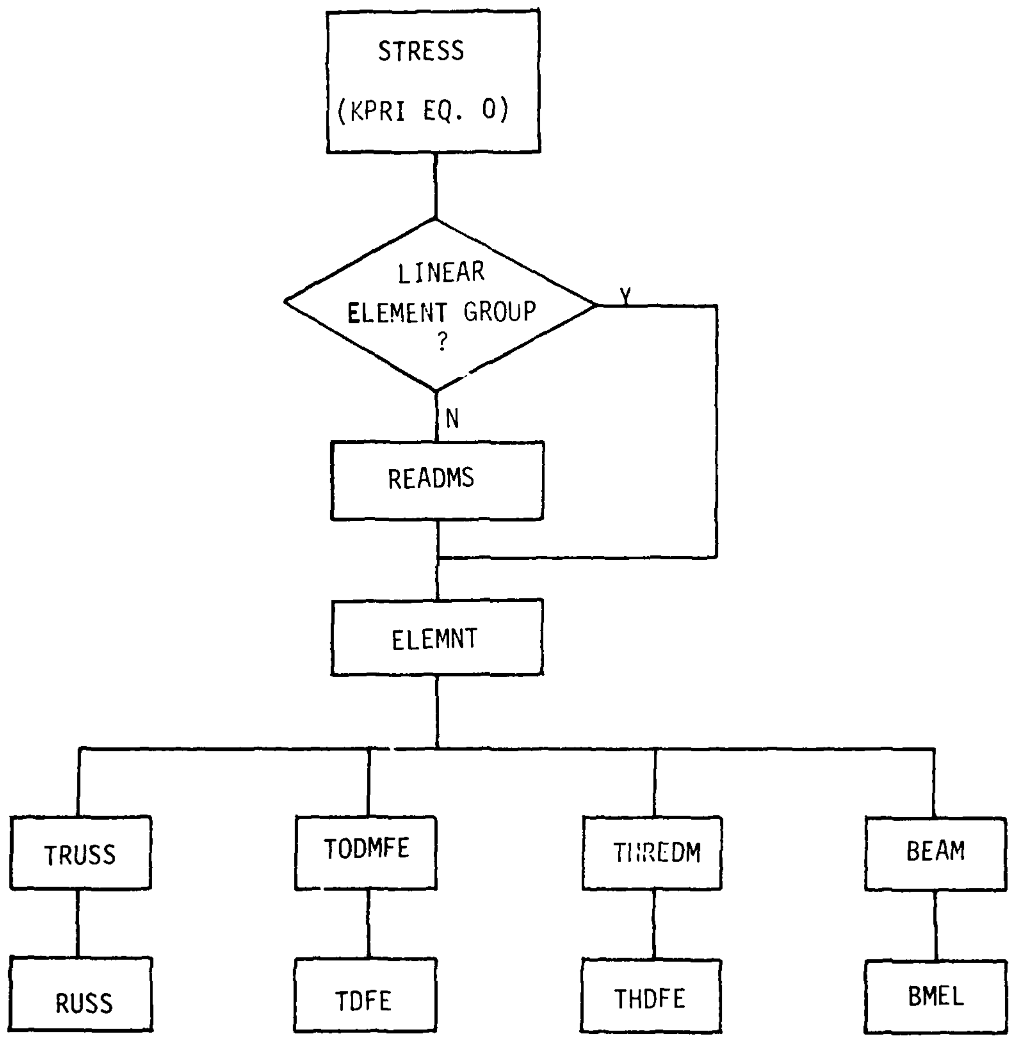
A6. Assembly of Structural Stiffness



A7. Equilibrium Iteration

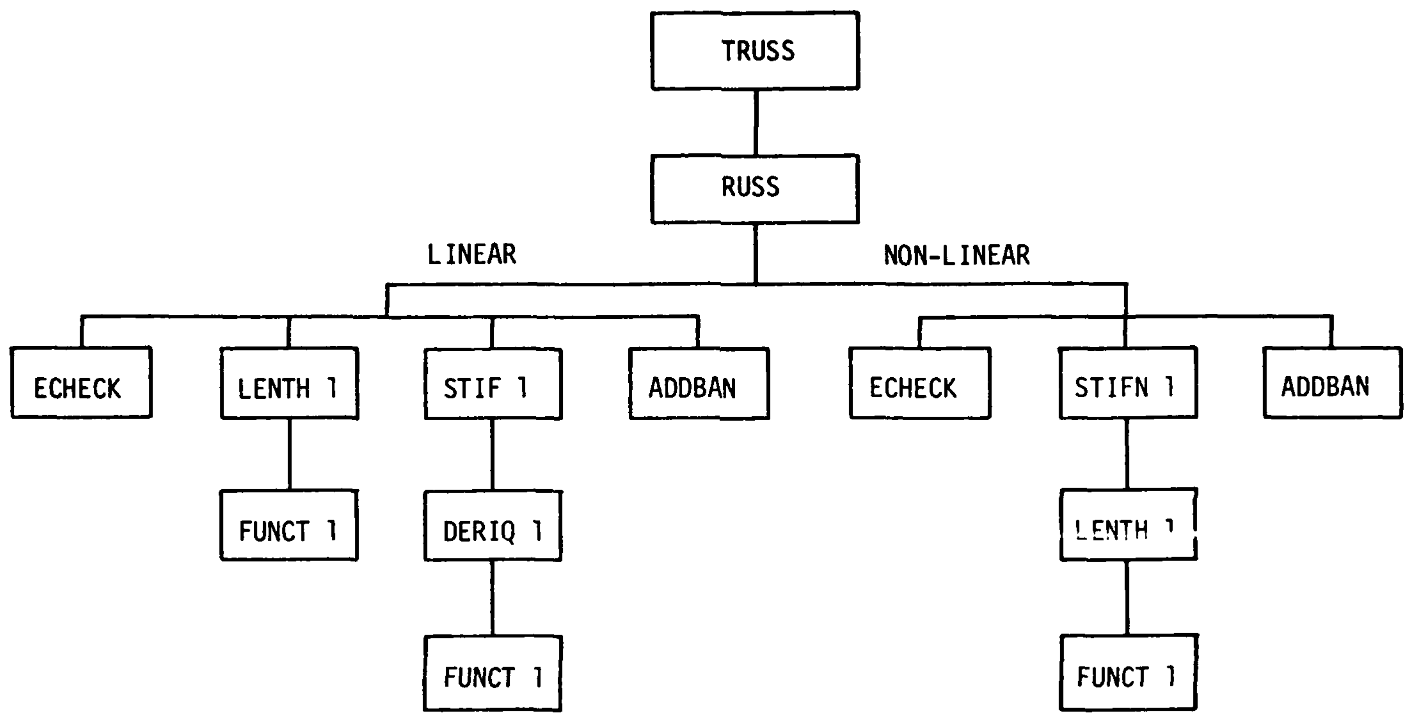


A8. Stress Calculation

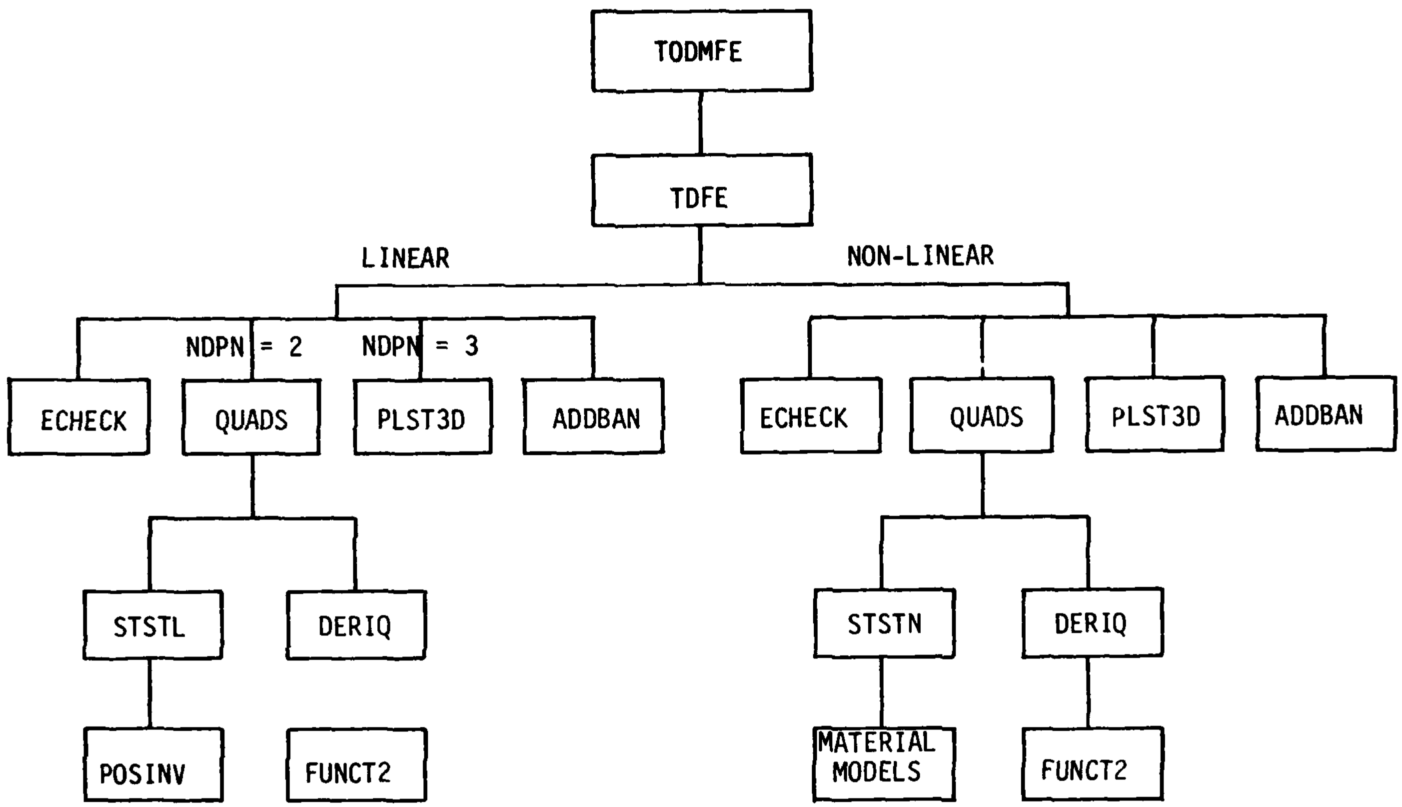


A17

A9. TRUSS ELEMENT



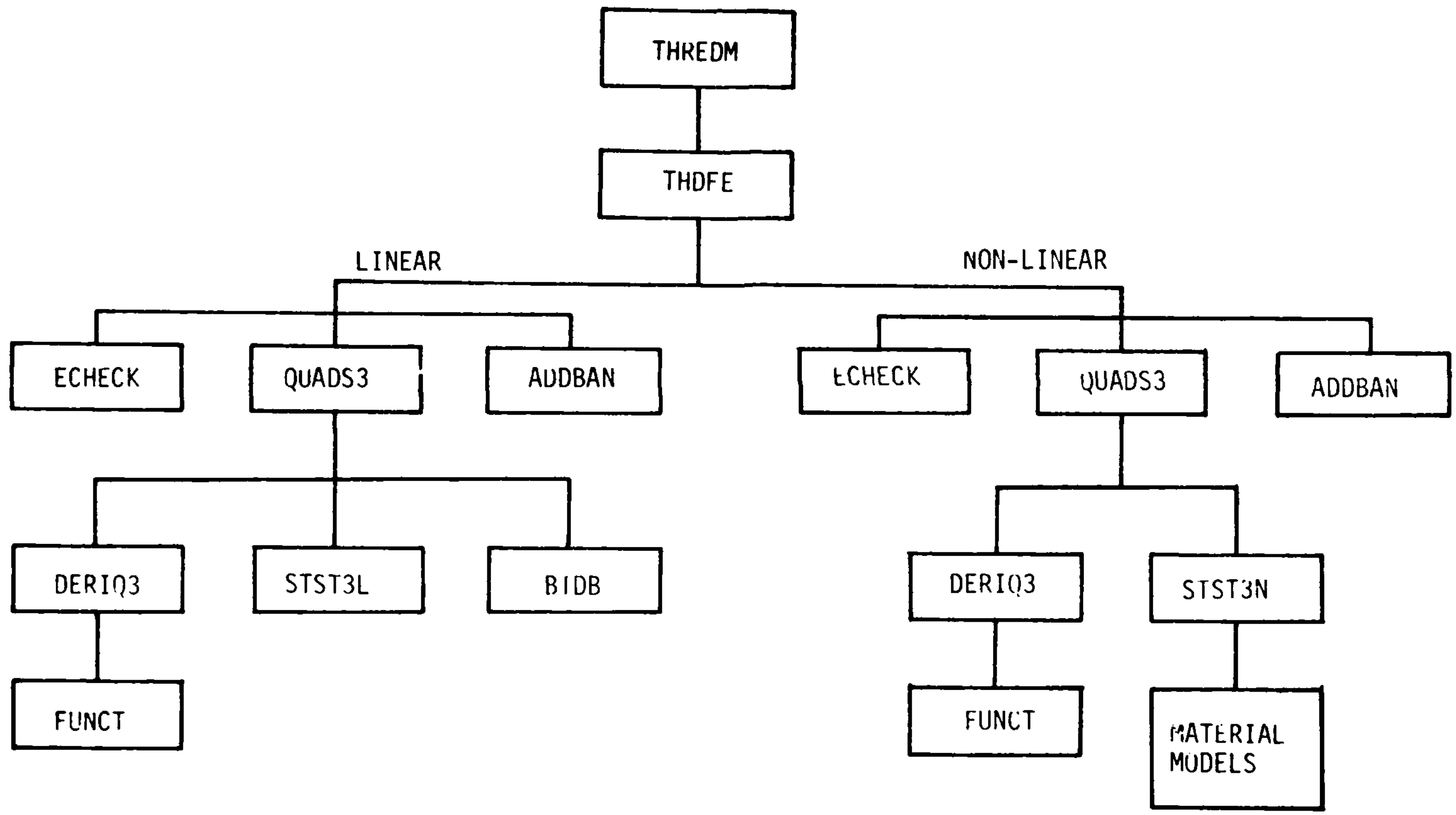
A10. <u>2-D Solid Element</u>



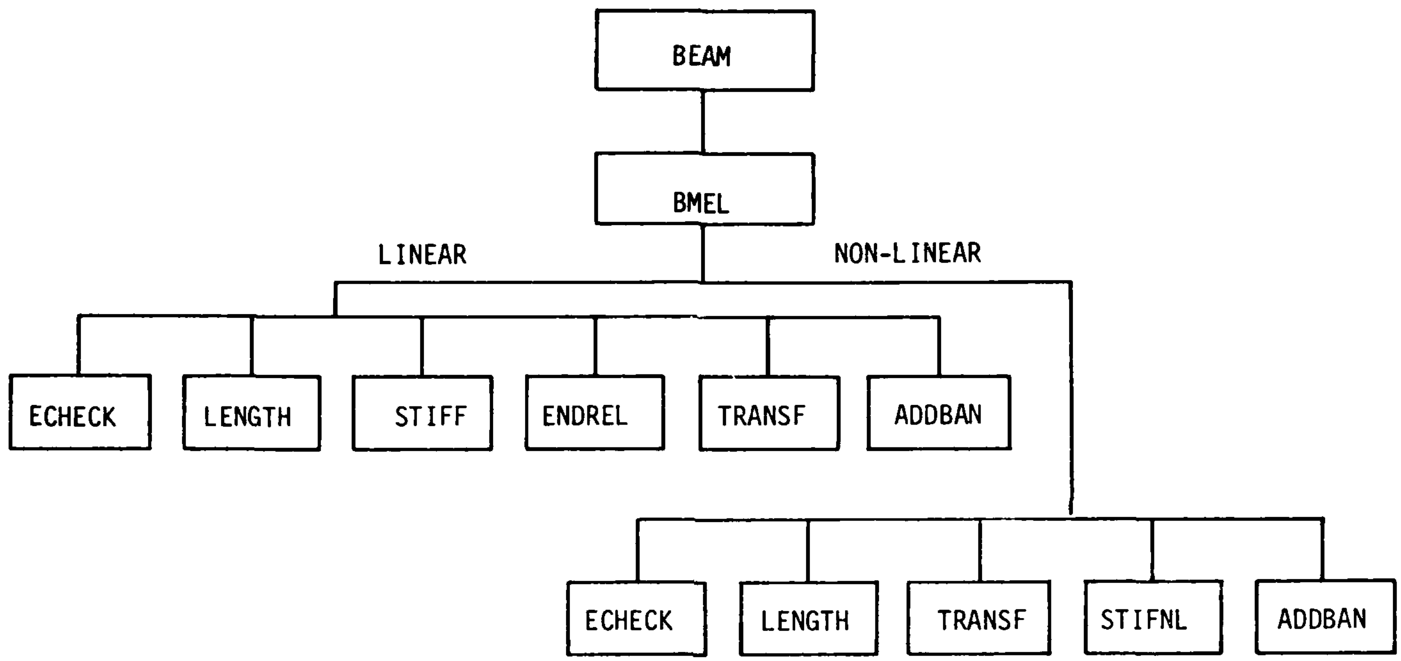
When NDPN EQ. 2, TWO degrees of freedom are assigned to each node.

When NDPN Eq. 3, THREE degrees of freedom are assigned to each node.

A11. 3/D Continuum Element



A12. Beam Element



Appendix B

DYNAMIC ALLOCATIONS OF ARRAYS IN THE BLANK COMMON BLOCK

B1. <u>Input Phase</u> - Nodal Data and Initialization of Arrays

Calling Subroutine: ADINI

Nodal point Coordinates:

Address	Dimension	INPUT	NODMAS	INITAL	Description
N1	NUMNP*NDOF	ID			Boundary condition array
N2	NUMNP*ITWO	X			Nodal point X-coordinate
N3	NUMNP*ITWO	Y			Nodal point Y-coordinate
N4	NUMNP*ITWO	Z			Nodal point Z-coordinate

Initial Conditions:

Address	Dimension	INPUT	NODMAS	INITAL	Description
N1	NUMNP*NDOF		ID	ID	Boundary condition array
N2	NEQ*ITWO			DISP	Displacement input
N3	NEQ*ITWO			VEL	Velocity input
N4	NEQ*ITWO			ACC	Acceleration input
N5	NEQ*ITWO		XMN[1]	DISPM[2]	1. Nodal mass vector or Nodal damping vector 2. For explicit time integration, the displacement vector at $t+\Delta t$.
N6					

N1 = NBCST +2*NEGNL +2
NBCST = Variable to shift start of BLANK COMMON beyond last address of program storage.
NEGNL = Number of nonlinear element group.

B2. Input Phase - Element Data

Calling Subroutines: TRUSS, TODMFE, THREDM, BEAM

Address	Dimension	TRUSS	TODMFE	THREDM	BEAM	Description
N1	NUMNP*NDOF	ID	ID	ID	ID	Boundary condition array
N2	NUMNP*ITWO	X	-	X	X	Nodal point X-coordinate
N3	NUMNP*ITWO	Y	Y	Y	Y	Nodal point Y-coordinate
N4	NUMNP*ITWO	Z	Z	Z	Z	Nodal point Z-coordinate
N5	NEQ+1	HT	HT	HT	HT	Column heights
N6A	(NUMNP+1)*ITWO	TEMPV1	TEMPV1	TEMPV1	TEMPV1	Nodal point temperatures
N6	MIDEST up to MAXEST	EE	EE	EE	EE	Element group information

Note: If temperature is not involved, N6 = N5 + NEQ.

B2-A. Element Data for Truss

Calling Subroutine: TRUSS

Address	Dimension	RUSS	Description
NFIRST	20	NPAR	Element group control parameters
N101 = N102	NUMMAT*ITWO	DEN	Mass density
N103	NUMMAT*ITWO	AREA	Cross-sectional area
N105	NDM*NUME	LM	Element connectivity array (vector)
N106	NDM*NUME*ITWO	XYZ	Element nodal coordinates
N107	NUME	MATP	Material property set number
N108	NUME*ITWO	EPSIN	Initial strain
N109	NUME	IPS	Stress printing flag
N110	MM*NUME*ITWO	ETIMV	Element birth/death time
N111	MM*NDM*NUME*ITWO	EDISB	Element disps. at birth time
N112	NUME*IDWA*ITWO	WA=IWA	Working array
N113	NCON*NUMMAT*ITWO	PROP	Material constants
N114	NDW*NUME*MXNODS	NODGL	Global node numbers
N115	NUME	IELTD	ELEMENT number of nodes
N116			

First Address: $\text{NFIRST} = \begin{cases} \text{N6} & \text{IND} \neq 4 \\ \text{N10} & \text{IND} \neq 4 \end{cases}$, $\text{MM} = \begin{cases} 0 & \text{IDEATH} = 0 \\ 1 & \text{IDEATH} = 1 \end{cases}$

Last Address: NLAST = N116

Total Length of Record: MIDEST = NLAST-NFIRST + 1

B2-B. Element for 2-D Element

Calling Subroutine: TODMFE

Address	Dimension	TYPE	Description
NFIRST	20	NPAR	Element group control parameters
N101	NDM*NUME	LM	Element connectivity array
N102	NDM*NUME*ITWO	YZ	Element nodal coordinates
N103	NUME	IELT	Element number of nodes
N104	NUME	IPST	Stress printing flag
N105	NUME*ITWO	BETA	Material angle
N106	NUME*ITWO	THICK	Element thickness
N107	NUME	MATP	Material property set number
N108	NUMMAT*ITWO	DEN	Mass density
N109	NCON*NUMMAT*ITWO	PROP	Material constants
N110	IDWA*NUME*ITWO	WA	Working array
N111	ND5DIM*NUME	NOD5	Midside nodes location array
N112	MM*NUME*ITWO	ETIMV	Element expiry time array
N113	MM*NUME*NDM*ITWO	EDISB	Element birthtime nodal coordinates
N114	9*NTABLE-1	ITABLE	Stress output location tables
NLAST			

B2-C. Element Data for 3-D Element

Calling Subroutine: THREDM

Address	Dimension	THDFE	Description
NFIRST	20	NPAR	Element group control parameters
N101	NDM*NUME	LM	Element connectivity array
N102	NDM*NUME*ITWO	XYZ	Element nodal coordinates
N103	NUME	IELTD	Element number of nodes for shape function
N104	NUME	IELTX	Number of nodes used to describe element geometry
N105	NUME	IPST	Stress printing flag
N107	NUME*ITWO	MATP	Material property set number
N108	ND9DIM*NUME	NOD9	Midside nodes location array
N109	NUME	IREUSE	Index for repeating element
N110	NUMMAT*ITWO	DEN	Mass density
N111	NCON*NUMMAT*ITWO	PROP	Material constants
N112	IDWA*NUME*ITWO	WA	Working array
N113	MM*NUME*ITWO	ETIMV	Element expiry time array
N114	MM*NUME*NDM*ITWO	EDISB	Element birthtime nodal coordinates
N115	16*NTABLE	ITABLE	Stress output location tables
N116	9*NORTHO*ITWO	DCA	Direction cosine array
N117	NUME	MAXESV	Material axis orientation storage vector
NLAST			

B2-C. Element Data for 3-D Element (Continued)

Calling Subroutine THREDM

Address	Dimension	THDFE	Description
NLAST	1	-	-
N120	NDM2*ITWO	S	Element stiffness matrix
N121	NDM*ITWO	XM	Nodel mass
N122	NDM*ITWO	B	Compacted strain-disp. matrix
N123	NDM*ITWO	RE	Element nodal force vector
N124	NDM*ITWO - 1	EDIS	Element displacement vector
N125			

B2-D. Element Data for BEAM

Calling Subroutine: BEAM

Address	Dimension	BMEL	Description
NFIRST	20	NPAR	Element group control parameters
N101	NMAT*ITWO	E	Young's modulus
N102	NMAT*ITWO	G	Shear modulus
N103	NUMMAT*ITWO	DEN	Mass density
N104	NMAT*ITWO	XI	Second moment of area @ 'R' axis
N105	NMAT*ITWO	YI	Second moment of area @ 'S' axis
N106	NMAT*ITWO	ZI	Second moment of area @ 'T' axis
N107	3*NMAT*ITWO	AREA	Normal + Shear section area
N108	9*NUME*ITWO	XYZ	Element coordinates array
N109	12*NUME	LM	Element connectivity array
N110	NUME	IPS	Stress output flag
N111	NUME	MATP	Material property set number
N112	NFAC*NCON*NUMMAT*ITWO	PROP	Material constants
N113	NFAC*NUMMAT	ICS	Section identification flag
N114	NFAC*NUMMAT	ISHEAR	Flag for transverse shear effects

B2-D. Element Data for BEAM (Continued)

Calling Subroutine: BEAM

Address	Dimension	BMEL	Description
N115	NFAC*IDWA*NUME*ITWO	WA	Working array
N116	LL*NFAC*NTABLE	ITABLE	Stress output
N117	INSR*NFAC*NUME*ITWO	SR	Gauss elimination coefficient
N118	MM*NUME*ITWO	ETIMV	Element expiry array
N119	12*MM*NUME*ITWO	EDISP	Element disps. at birth time
N120	6*NMOMNT	IMOMNT	Number of end release tables
N121	NUME	IELRET	Element number of end release table
N122	12*NUME*ITWO	PDISP	Element displacement vector
N123	NUME*ITWO	GAMA	Creates element transformation matrix
NLAST			

B3. Addresses of Diagonal Elements of Structural Stiffness Matrix

Calling Subroutine: ADDRES

Address	Dimension	Address	Description
N1	NDOF*NUMNP	MAXA	Locations of diagonals of structural stiffness matrix
N2			
	. . .		
	. . .		
N5	NEQ	MHT	Column height
N6			

Note: If temperature is involved, N6 = N5 + NEQ + 1.

B4. Determination of Blocks for Stiffness Matrix

Calling Subroutine: SBLOCK

Address	Dimension	SBLOCK	Description
N1	NEQ + 1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NEQ	NCOLBV	Number of columns per block
N1B	NBLOCK	ICOPL	First coupling block

B5. Assembly of Constant Structural Matrices

Calling Subroutine: ASSEM

Address	Dimension	ASSEM	Description
N1[1]	NEQ + 1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK	ICOPL	First coupling blocks
N1C	NEGNL*NBLOCK	IGRBLC	Nonlinear element group to stiffness block coupling
N1D[2]	(IEIG +1)*NBLOCK+1		For the case of frequency analysis
N2	ISTOH*ITWO	AA	Storing blocks of matrices
N3	ISTOH*ITWO	CC	Storing blocks of matrices
N4[3]	NEQ*ITWO	DD	Concentrated masses and lumped mass matrix
N5	NEQ*ITWO	BB	Concentrated dampers
N6[4]	MAXEST+NBCEL	EE	Element group storage
N7			(Storage for element calculations)

Notes:

1. N1 = NBCST+2*NEGNL+2
2. For one block case, N1D = N1C
3. For implicit time integration and one block case N4 = N3
   For explicit time integration N4 = N1
4. For static analysis N6 = N4
   For static analysis with lumped gravity load N6 = N5.

B6. Construction of Load Vectors

Calling Subroutines: LOAD, CLOADS, GRAVL, TODPRL, THDPRL

Address	Dimension	Arrays	Description
N1	NEQ + 1	MAXA	Locations of diagonals of structural stiffness matrix
N2	NODF*NUMNP	ID	Boundary condition array
M2	NTFN*NSTE*ITWO	RG	Interpolated values of time functions
M3	NTFN*ITWO	RGST	Initial values of time functions
M4	NEQ*ITWO	R	Load vector
MM			

Notes:

M4 plus	(2*NPTM + 2*NLOAD + NEQ)*ITWO + (NEQ + 4*NLOAD)	in calculation of concentrated load vector
M4 plus	(3*NUMNP + NDFR2*NPR2 + 4*NPR2)*ITWO + NPR2*(NDFR2 + 6)	in calculation of 2/D pressure load vector
M4 plus	(3*NUMNP + NDFR3 + 5*NPR3)*ITWO + NPR3*(NDFR3 + NODE3 + 2)	in calculation of 3/D pressure load vector

B7. Transfer of Data for a Restart Job

Calling Subroutine: RSTART

Address	Dimension	RSTART	Description
N2	NEQ*ITWO	DISP	Initial displacement
N7	NEQ*ITWO	VEL	Initial velocity
N8	NEQ*ITWO	ACC	Initial acceleration
N10	MAXEST	EE	Element information

B8. Calculation of Effective Load Vector

Calling Subroutine: LOADEF

Address	Dimension	LOADEF	Description
N1	NEQ + 1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK		
N1C	NBLOCK*NEGNL		
N1D[1]	(IEIG+1)*NBLOCK+1		
N2[2]	NEQ*ITWO	DISP	Initial displacement
N3	NEQ*ITWO	R	Load vector
N4	ISTOH*ITWO	AA	Linear part of structural stiffness
N4A	ISTOH*ITWO		
N4B[1]	NEQ*ITWO		
N5[2]	NEQ*ITWO		
N6	NEQ*ITWO	WV	Working vector
N6A	(ITEMPR-1)*(NUMNP+1)*ITWO		
N6B	(NUMNP+1)*ITWO		
N7	NEQ*ITWO	VEL	Nodal velocity
N8	NEQ*ITWO	ACC	Nodal acceleration
N9	NEQ*ITWO	XM	Nodal mass
N10			

Notes: 1. For one block case, N1D=N1C, N4B = N4A

2. For explicit time integration N2 = N1 + NEQ*ITWO, N5 = N4

B9. Assembly of Tangent Structural Stiffness

Calling Subroutine: ASSEM

Address	Dimension	ASSEM	Description
N1	NEQ+1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK		
N1C	NBLOCK*NEGNL	IGRBLC	Nonlinear element group to stiffness block coupling
N1D	(IEIG+1)*NBLOCK+1		
N2	NEQ*ITWO	DD	Concentrated masses and lumped mass matrix
N3	NEQ*ITWO	BB	Concentrated dampers
N4	ISTOH*ITWO	AA	Storing blocks of matrices
N4A	ISTOH*ITWO	CC	Storing blocks of matrices
N4B	NEQ*ITWO		
N5	NEQ*ITWO		
N6	NEQ*ITWO		
N6A	(ITEMPR-1)*(NUMNP+1)*ITWO		
N6B	(NUMNP+1)*ITWO	TEMPV2	Nodal point temperatures
N7	NEQ*ITWO		
N8	NEQ*ITWO		
N9	NEQ*ITWO		
N10	MAXEST+NBCEL	EE	Element group storage
N11			

B10. Linear Equation Solver

Calling Subroutine - COLSOL

Address	Dimension	COLSOL	Description
N1	NEQ+1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK	ICOPL	First coupling block
N1C	NBLOCK * NEGNL		
N1D	(IEIG+1)*NBLOCK+1		
N2	NEQ*ITWO		
N3	NEQ*ITWO	V	Load or displacement vector
N4	ISTOH*ITWO	A	Structural stiffness matrix
N4A	ISTOH*ITWO	B	Working space for blocked structural stiffness
N4B	NEQ*ITWO	D	Reduced diagonal location
N5			

B11. Equilibrium Iteration

Calling Subroutine: EQUIT

Address	Dimension	EQUIT	Description
N1	NEQ + 1	MAXA	Addresses of diagonal elements in effective stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK	ICOPL	First coupling block
N1C	NBLOCK*NEGLNL		
N1D	(IEIG+1)*NBLOCK+1		
N2	NEQ*ITWO	DISP	Displacement at previous time step
N3	NEQ*ITWO	DISPI	Displacement increment at previous time step
N4	ISTOH*ITWO	AA	Effective stiffness matrix and working storage
N4A	ISTOH*ITWO	CC	Working storage in out-of-core solution
N4B	NEQ*ITWO	DK	Elements of D in LDL^T factorization of effective stiffness matrix
N5	NEQ*ITWO	RE	Out of balance loads

B11. Equilibrium Iteration

Calling Subroutine: EQUIT

Address	Dimension	EQUIT	Description
N6	NEQ*ITWO	WV	Working Array
N6A	(ITEMPR-1)* (NUMNP+1)*ITWO		
N6B	(NUMNP+1)*ITWO		
N7	NEQ*ITWO	VEL	Velocity at previous time step
N8	NEQ*ITWO	ACC	Acceleration at previous time step
N9	NEQ*ITWO	XM	Nodal mass
N10	MAXEST+NBCEL	EE	Element group storage

B12. Storage Allocation During Static Analysis OR Implicit Time Integration

Address	Dimension	Arrays	Description
N1	NEQ+1	MAXA	Locations of diagonals of structural stiffness matrix
N1A	NBLOCK	NCOLBV	Number of columns per block
N1B	NBLOCK	ICOPL	First coupling block
N1C	NEGNL*NBLOCK	IGRBLC	Nonlinear element group to stiffness block coupling
N2	NEQ*ITWO	DISP	Displacement vector
N3	NEQ*ITWO	R	Load vector and displacement increments
N4	ISTOH*ITWO	A	
N4A	ISTOH*ITWO	B	Storage for solution of equations
N4B	NEQ*ITWO	Dk	
N5	NEQ*ITWO	RE	Our-of balance loads, and correction to displacement increment
N6	NEQ*ITWO	WV	Working vector
N6A	(NUMNP+1)*ITWO	TEMPV1	Working vectors to share nodal point temperatures
N6B	(NUMNP+1)*ITWO	TEMPV2	
N7	NEQ*ITWO	VEL	Velocity vector for dynamic analysis only
N8	NEQ*ITWO	ACC	Acceleration vector for dynamic analysis only
N9	NEQ*ITWO	XM	Lumped mass matrix for dynamic analysis only
N10	MAXEST	EE	Element group information
	NBCEL		
Total			

B13. Storage Allocation During Explicit Time Integration

Address	Dimension	Arrays	Description
N1	NEQ*ITWO	DISPM	Displacement vector at time t - Δt
N2	NEQ*ITWO	DISP	Displacement vector at time t
N3	NEQ*ITWO	R	Load vector at time / or displacement vector at time t + Δt
N4 = N4A = N4B = N5 = N6	NEQ*ITWO	WV	Working vector
N6A	(NUMNP+1)*ITWO	TEMPV1	Working vectors to store nodal point temperatures
N6B	(NUMNP+1)*ITWO	TEMPV2	
N7	NEQ*ITWO	VEL	Velocity vector at time t
N8	NEQ*ITWO	ACC	Acceleration vector at time t
N9 = N10	MAXEST+NBCEL	EE	Element group information
Total			

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AUE-801 ✓	2. GOVT ACCESSION NO. AD-A096678	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Education of ADINA: Part I - Theory and Programing Description.		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) T. Y. Chang, J. Padovan		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0691
9. PERFORMING ORGANIZATION NAME AND ADDRESS College of Engineering University of Akron Akron, OH 44325		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA		12. REPORT DATE June 1980 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ADINA, Nonlinear Finite Element Analysis, Computer Program

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An evaluation of 1977 ADINA, a general purpose nonlinear finite element Program, was conducted. The evaluation work consists of the review of its theoretical basis, nonlinear static and dynamic solution algorithms, and program architecture. A discussion of the program is also made with respect to its nonlinear analysis capability and limitations.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DATE
ILMED
-8